ANNO 2 - NUMERO 5 FEBB./MAR. 1989 L. 14.000 - Frs. 21.00

GRUPPO EDITORIALE JACKSON
AREA CONSUMER

PASSE-PARTOUT NELL'UNIVERSO DI AMIGA

CONTIENE AMIGA Disk Magazine

Scacchi

Amiga Fotogenico

MUSICA

L'INTERFACCIA MIDI PER AMIGA

LE PERFORMANCE MUSICALI DI AMIGA

Mauro Graziani

TOP GAME: GROWTH

Modula-2

INTUITION

RUBRICHE:

MERCATINO, VETRINA, SCORCIATOIA

CLASSIFICHE, FUMETTI IN COMPUTER -GRAFICA

Spedizione in abbonamento postale Gruppo III/70 Taxe percue (Tassa riscossa) Milano Ferr. Corr.

EDITORIALE

E' successo, da questo numero il tanto famigerato D.T.P è entrato prepotentemente nella nostra rivista. Purtroppo, però, i possessori di Amiga non possono ancora beneficiare di tali tecniche, essendo la nostra amata macchina ancora, inspiegabilmente, a tal fine sprovvista di validi supporti, sia software che hardware. I pochi sistemi di D.T.P. finora visti per tale macchina, infatti, non hanno certo le caratteristiche di affidabilità e di professionalità indispensabile per poter operare in questo non facilissimo ambito. A tale proposito, chiediamo fin da ora scusa per eventuali sviste che potrebbero comparire qua e là nella rivista: non dovrebbero essercene poi molte più del solito.

Asciugandoci le lacrime per l'ennesima occasione persa dal mondo Amiga (mi riferivo ancora al D.T.P.), veniamo alle novità di questo mese.

Come prima leccornia segnalerei senz'altro l'inizio di alcuni articoli dedicati al Modula-2, che penso faranno la gioia di tutti coloro i quali conoscono il linguaggio Pascal e vista, come al solito, l'impossibilità di lavorare con il turbo su Amiga si siano decisi a passare ad un linguaggio dotato di più verve. Ovviamente spero che il succitato ciclo di articoli faccia la "felicità" anche di chi, pur non conoscendo il Pascal, voglia avvicinarsi alla programmazione strutturata per tramite di un linguaggio non di basso livello.

Sempre per chi ama la programmazione, consiglio l'articolo di Antonio Ferrillo La gestione di Intuition, *dove viene spiegato dettagliatamente come provvedere allo sviluppo di una libreria di funzioni personali, per poter, appunto, gestire Intuition.*

Per i musicofili, invece, siamo lieti di poter pubblicare un'intervista a Mauro Graziani -nome di certo non nuovo a chi conosca lo scenario nazionale ed internazionale della musica elettronica-, dove nonostante non si parli di Amiga, dopotutto non è mica sempre indispensabile, vengono affrontate molte delle problematiche inerenti al fare musica, o per meglio dire del fare musica elettronica.

Come non bastasse, sempre per gli amanti del pentagramma, un articolo sulle performance al SIM e allo SMAU di un gruppo che invece fa musica elettronica proprio con Amiga e che pertanto ha molto da dire e da consigliare in proposito. Continua anche la serie di articoli dedicati alle interfacce MIDI.

Per ora vi saluto, non senza prima ricordarvi il meraviglioso TOP GAME del mese: GROWTH.

Arrivederci al prossimo numero.

SOMMARIO

Foto di copertina realizzata
nellla redazione grafica della:
Graphic & Comp **di Gorizia**

Anno 2 numero 5 Febbraio/Marzo 1989

DIRETTORE RESPONSABILE
Paolo Reina

REDAZIONE
Graphic & Comp. Gorizia

COORDINAMENTO REDAZIONALE
Simone Concina

ART DIRECTOR
Gianni Marega

COLLABORATORI
Roberto Beccia, Primoz Beltram, Tomi Beltram, Fabio Biancotto, Giorgio Dose, Mr. Lambda, Massimo Lavarin, Furio Lusnig Luigi Manzo, Giovanni Michelon, Emilio Orione, Alessandro Prandi, Giacomo Pueroni, Paolo Russo.

GRAFICA, IMPAGINAZIONE ELETTRONICA E COPERTINA
Graphic & Comp.

REVISIONE GRAFICA E IMPAGINAZIONE
Gianni De Tomasi

DIVISIONE PUBBLICITÀ
Via Pola 9 - 20124 MILANO - Tel. 69.481
Telex 316213 REINAI - 333436 GEJ - ITI
OVERSEAS DEPARTMENT: Tel. 02/6948492
PUBBLICITÀ PER ROMA-LAZIO E CENTRO SUD
Via Lago di Tana, 16 - 00199 Roma
Tel. (06) 8380547 - Telefax (06) 6948438

STAMPA
Grafiche FBM - Gorgonzola (MI)

DISTRIBUZIONE
Sodip - Via Zuretti, 25 - 20125 MILANO
Spedizione in abbonamento postale Gruppo III/70

UFFICIO ABBONAMENTI
Tel. (02) 6122527-6187376
Prezzo della rivista L. 14.000 (Frs. 21.00)
Numero arretrato L. 28.000
Abbonamento annuo L. 92.500
per l'Estero L. 185.000
I versamenti vanno indirizzati a:
Gruppo Editoriale Jackson
Via Rosellini, 12 - 20124 Milano
mediante emissione di assegno bancario, vaglia o utilizzando il C/C/ postale numero 11666203
Per i cambi di indirizzo, indicare, oltre al nuovo, anche l'indirizzo precedente.

PUBLISHER
Filippo Canavese

GRUPPO EDITORIALE JACKSON
AREA CONSUMER

DIREZIONE
REDAZIONE, AMMINISTRAZIONE
Via Rosellini, 12 - 20124 Milano
Tel. (02) 66800000-66800161 - Telex 333436 GEJIT I
SEDE LEGALE
Via Pietro Mascagni, 14 - 20122 Milano
Il Gruppo Editoriale Jackson è iscritto nel Registro nazionale della Stampa al n. 117 vol. 2 - foglio 129 in data 17/8/1982

Autorizzazione del Tribunale di Milano n. 102 del 22/2/1988

Piu' colori al desktop

Shakespeare, package di DTP per la famiglia Amiga, prodotto dalla Infinity recentemente ha aggiornato il programma con una nuova versione, la quale permette l'integrazione tra testo e grafica sopportando piu' immagini a colori (De Luxe Paint) e più palette di colori su una stessa pagina. Shakespeare può effettuare l'output alle device PostScript, incluse l'Apple Laserwriter Plus e la Linotronic Imagesetter. Il prezzo sul mercato inglese e' di Lire Sterline 148.35.

Da Commodore a PS/2

Gli ingegneri della Commodore in Germania hanno stabilito che l'architettura del Microchannel del PS/2 dell'IBM è abbastanza vicina a quella dell'Amiga 2000 per essere semplicemente inclusa in quest'ultimo.

Questo significa che oltre alle bridgeboard XT e AT presto potremmo avere a disposizione una scheda PS/2 con VGA e Microchannel.

Nuove da casa Aegis

La casa californiana di software sta lanciando un'offensiva commerciale destinata a dare i suoi frutti, perlomeno alle apparenze. In passato i package grafici di questa casa non hanno avuto molta fortuna, vedi Aegis Draw, di gran lunga inferiore a De Luxe Paint, d'altra parte le regole del mercato sono inesorabili, per cui l'Aegis ha dovuto reimpostare i suoi piani per quanto riguarda i nuovi prodotti che, stando alle voci di corridoio, sembrano di buona qualità. Il nuovo progetto dovrebbe chiamarsi 'Videola!' che sta per Video-Editing-Software. Dall'animazione ad altri innumerevoli effetti grafici sono in fase di ideazione e attuazione alle dipendenze del signor Gary Bonham, programmatore occasionale dell'Aegis.

Regolamento del concorso

Il regolamento del concorso è articolato in dieci punti.

1 - Il Circolo Arci Rataplan in collaborazione con il Comune di Riccione, organizza il secondo concorso per animazione grafica in tempo reale.

2 - Sono ammesse tutte le opere realizzate su personal computer ed elaborate in tempo reale.

3 - Possono essere utilizzati tutti i programmi grafici esistenti, ma non oggetti o sfondi tratti da librerie di pubblico dominio.

4 - Ogni concorrente può presentare al massimo quattro opere.

5 - I lavori dovranno essere registrati su floppy disk o hard disk.

6 - Le opere dovranno pervenire, franco ogni spesa, al seguente indirizzo: Circolo Arci Rataplan c/o Mainardi Carlo - Via Bologna n.13 - 47036 Riccione (FO).

Queste dovranno essere accompagnate da una scheda di commento.

7 - Le opere verranno preventivamente esaminate da una commissione che deciderà se ammetterle al concorso.

8 - I lavori dovranno pervenire entro il giorno 15 marzo 1989.

9 - Le opere saranno giudicate dal pubblico con apposite schede de voto e da una giuria qualificata.

10 - Il supporto magnetico sarà restituito all'autore al termine del concorso.

Ai primi tre classificati verranno consegnati dei premi in denaro. Per ulteriori informazioni rivolgersi a Carlo Mainardi (tel. 0541-42878) oppure a Stefano Leardini (tel. 0541-377388).

Bit movie '89

Nel periodo pasquale del marzo 1989, si terrà a Riccione la seconda edizione di BIT MOVIE. La mostra, visto il successo ottenuto nello scorso anno, conserverà il suo carattere originale, riservando la sezione concorso relativa all'animazione computerizzata in tempo reale a lavori eseguiti su macchine appartenenti alla fascia dei personal computer. La sezione concorso costituisce il centro d'interesse dell'intera manifestazione. La mostra è suddivisa in ulteriori cinque sezioni.

Compito principale della sezione "La prossima rivoluzione nei computers: dal personal al supercomputer, il parallel processing e i computers della quinta generazione" è quello di fornire un'adeguata informazione scientifica sulle architetture a processori paralleli e sulle possibilità dei nuovi media di diventare una "universal intellectual utility".

La sezione laboratorio si chiamerà "Desktop video! using Amiga video tools" ed ospiterà un laboratorio di video post produzione basato sul computer Amiga. Una delle sezioni ospiterà alcuni esempi di possibili applicazioni del mezzo elettronico alla città come strumento informativo o come mezzo divulgativo: guida elettronica ai monumenti della città di Rimini, catalogo elettronico multilingue del nuovo museo di Riccione. La sezione video ospiterà le più recenti opere realizzate su supercomputer e presentate al Festival Internazionale di Montecarlo. La sezione musicale è stata ampliata e tratterà varie tematiche: Computer ed educazione musicale; computer e strumenti musicali; la gestione degli strumenti tramite computer; computer, midi e strumenti: la sala di incisione e la one man band.

Gentilissima Redazione, ho aspettato di avere il secondo numero della vostra rivista "Amiga Magazine" per avere la conferma delle prime impressioni avute durante la lettura del primo numero e devo ammettere che sono state confermate. Il risultato è: congratulazioni a tutto lo staff che collabora alla rivista. Comunque la casa editrice non si smentisce, poiché compro altre riviste da Voi edite.

Mi presento. Sono un possessore di Commodore, 64 e 500, con una vasta gamma di periferiche connesse, che sono solo in aumento. Sono un militare ed ho 39 anni, è dal 1982 che compro computer (ho iniziato con lo Spectrum), non ho mai frequentato corsi di informatica, tutte le mie conoscenze sono autodidattiche grazie ai libri da Voi editi per computer specifici: ad esempio ho comperato i tre volumi sull'Amiga da Voi pubblicati, ed ho apprezzato moltissimo le spiegazioni date dalla Signora Rita Bonelli sul volume "Amiga 500".

Ed ora vi faccio la richiesta di prammatica. Poiché ho deciso di seguire e studiare le vostre lezioni sul linguaggio "C" e sul linguaggio "Assembler", avrei un suggerimento da dare che spero accogliate. So per esperienza mia personale che una volta arrivati ad un certo numero di "uscite", e Voi avete intenzione di continuare per parecchio tempo visto che avete iniziato con "Anno I - Numero 1", le riviste, per ragioni di spazio in biblioteca dove i libri hanno la precedenza, vengono poi confinate in posti poco pratici per una eventuale "ripassata" di argomenti poco capiti inizialmente.

Vi chiedo se potete fare in modo di avere questi articoli impaginati in modo tale da poterli poi mettere insieme; anche se questa soluzione mi dispiacerebbe perché mi costringerebbe a rompere la rivista. Allora l'altra soluzione potrebbe essere che Voi metteste i testi sul dischetto che è allegato alla rivista. Attualmente io ricopio tutti gli argomenti che si prevede abbiano un seguito lungo, 4 o 5 numeri, con un word processor, e Vi confesso che mi è molto faticoso e occupa molto del tempo prezioso che potrei invece dedicare al computer: faticoso perché sono uno dei tanti che usa solo due dita per battere; dispendioso perché è il tempo dedicato all'hobby e ognuno di noi sa quanto poco tempo gli si può dedicare.

Ed ora la richiesta. potete pubblicare un articolo, come quello apprezzato su Commodore Professional n.2 "Anatomia di un disco", che analizzi e spieghi il boot dell'Amiga? Tutto qui, La ringrazio dell'attenzione prestatami e spero di non essere stato troppo esigente.

Le porgo i miei più cordiali saluti e un augurio per il futuro.

Corrado Giannella
Sqd. Italair
Nagoura (Libano)

Egregio Sig. Giannella, innanzi tutto un grazie sentito per i gentili complimenti, i suggerimenti da lei dati sono senz'altro di notevole interesse e, a ben vedere, di estrema utilità per l'utenza. Ciò che però non ha tenuto in considerazione, è che lo spazio disponibile sul dischetto allegato è ridottissimo se non addirittura nullo. Ci è pertanto impossibile esaudire le sue richieste, lo si potrebbe eventualmente fare tramite una rivista completamente ospitata su dischetto; ma questo è già un altro discorso.

Le spiegazioni relative al funzionamento del dischetto in Amiga, saranno senz'altro argomento di prossima pubblicazione

Spett.le Redazione, ho acquistato in edicola la rivista Amiga Magazine di settembre - ottobre e sono rimasto particolarmente interessato dall'articolo sull'interpolazione (Spline) di Giovanni Michelon e Luigi Manzo.

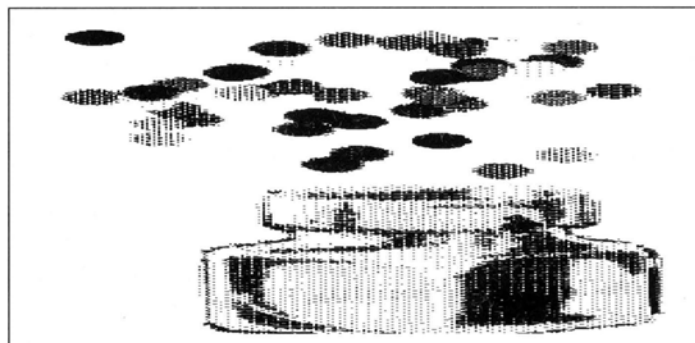
Pur essendo un utente MS-DOS sono riuscito ad utilizzare il programma INTERPOLA-2 descritto nella rivista, tuttavia nell'articolo menzionato non sono riuscito a trovare il programma B-SPLINE.

Poiché il problema mi interessa, vorrei sapere se è possibile averne il listato e se esiste la possibilità di interpellare direttamente gli autori dell'articolo.

Ringraziando per la gentile attenzione, Vi porgo i miei migliori saluti.

Alessandro Zanzi
Varese

Chiediamo scusa, ma il listato da lei citato è stato pubblicato sul dischetto che accompagnava il numero successivo a quello da lei citato.

Le sarà quindi sufficiente ordinarne una copia alla nostra sede per poterne entrare in possesso.

Avere la Jackson Card è semplice. La Jackson iniziative editoriali, infatti, ha creato una Gold Card per tutti gli abbonati e una Silver Card per tutti coloro che acquistano libri o riviste Jackson. E avere la Jackson Card significa tanti vantaggi tutti esclusivi: sconto sui corsi di formazione organizzati dalla Scuola di Alte Tecnologie Applicate Jackson S.A.T.A., gratis sei numeri di una delle riviste settimanali Jackson "Meccanica Oggi", "Informatica Oggi", "EO News settimanale di elettronica", gratis l'invio, personale e riservato, dei cataloghi libri Jackson e del mensile Jackson Preview Magazine. Tutto questo non è poco, ma da oggi non è tutto.

GRUPPO EDITORIALE JACKSON

tanto con tanto poco.

JACKSON GOLD CARD
GRUPPO EDITORIALE JACKSON

JACKSON SILVER CARD
GRUPPO EDITORIALE JACKSON

Potere

Avere Jackson Card significa poter acquistare nei migliori negozi di tutta Italia e spendere veramente meno. Infatti ogni titolare Jackson Card può acquistare tutti gli articoli messi in vendita nei negozi convenzionati Jackson godendo di sconti speciali.

Essere titolari Jackson Card perciò significa poter acquistare le cose più belle, finalmente a prezzi vantaggiosissimi.

GRUPPO EDITORIALE JACKSON

tanto con tanto poco.

'File.INFO che cosa sono?

Seppure normalmente, operando con Amiga, usate il Workbench e vi portate nel CLI solo occasionalmente, forse avrete notato che da CLI si vede un gruppo di file con gli stessi nomi di quelli contenuti sul disco solo con appeso al loro nome il suffisso '.infò. Questi sono i file che contengono le informazioni necessarie per operare nell'ambiente Workbench; tali informazioni servono all'Amiga per distinguere tra i vari tipi di file (Disco, Drawer, Tool, Project o Garbage) e per conoscere la situazione di cancellazione (file protetto o cancellabile - Protected/Deletable). Nel caso dei Tool si comunicano la grandezza dello Stack e i settaggi di default del programma selezionato. Se il file è un Project, comprenderà l'informazione riguardante il tool di default, un commento e ciascun tipo specifico di tool.

Per capire meglio di cosa stiamo parlando, selezionate un icona nel Workbench premendo un'unica volta il tasto sinistro del mouse, quindi scegliete l'opzione INFO dal menu Workbench. Tutti i dati relativi a quel file verranno visualizzati nella schermata INFO. Di tutti i termini, l'unico che possiede l'opzione On/Off è quello di Protezione/Sprotezione. Selezionando tale voce si passerà dalla posizione di Deletable a quella di Protected. Le altre campiture che si notano sono quelle riservate all'immissione di una stringa, nella quale è possibile inserire delle informazioni numeriche o di testo. Nel caso del gadget TOOL TYPES avete ben quattro campiture aggiuntive. Queste sono settate in modo da permettervi di aggiungere delle informazioni che verranno inviate al programma appropriato per settare a sua volta i default da voi selezionati. Nel caso di un Wordprocessor, ad esempio, i file testo vengono salvati di solito con delle indicazioni specifiche riguardanti il tipo di carattere usato.

Se provate a curiosare tra gli INFO dei file salvati con NotePad noterete che il font di default usato dal programma è il TOPAZ. 8, per cui i file vengono salvati con questo tipo di font. Se invece volete che il file venga scritto con i caratteri OPAL, non dovrete far altro che seguire i seguenti passaggi:

1> Selezionare l'icona del file premendo una volta sola il tasto sinistro del file.
2> Scegliere l'opzione INFO dal menu Workbench.
3> Clickare il gadget ADD nella campitura Tool Type e quindi premere una volta il tasto sinistro del mouse posizionando il cursore nell'area riservata al testo.
4> Scrivete "FONT=opal. 9" è premete <Return>.
5> Selezionate il gadget SAVE.

Ora, ogniqualvolta selezionerete il file con questo INFO il Workbench caricherà NotePad e il font usato sarà OPAL. 9.

Creazione e copiatura di Directory da Workbench

Di sicuro esaminando il dischetto del Workbench, avrete notato una icona chiamata 'Empty' (vuoto) che, senza un motivo apparentemente valido, si presenta nella window principale del dischetto. Ebbene il motivo c'è ed è più che valido. Se non avete ancora molta dimestichezza con i comandi CLI potete creare delle directory su un vostro dischetto formattato direttamente da Workbench selezionando l'icona Empty e spostandola nella finestra del vostro disco, a questo punto scegliete l'opzione 'Renamè dal menu del Workbench e rinominate la directory come vi sembra più opportuno. Per molti di voi senz'altro si tratta della scoperta dell'acqua calda, ma visto che questa informazione non è riportata su nessun manuale ci è sembrato utile aiutare i nuovi possessori di Amiga.

Altri aspetti del comando DIR

Il comando Dir del CLI possiede delle caratteristiche particolari che nel manuale AmigaDOS non vengono riportate. Al comando Dir si può aggiungere un espressione simile a quella usata per il PAT nei comandi List, in tal modo verranno visualizzati tutti i files nelle directory con un predeterminato pattern. Se per esempio il comando 'dir ????' visualizza tutti i file contenuti nelle directory con il nome di quattro lettere, 'dir c#?' mostrerà tutti i file all'interno delle directory che iniziano con la lettera 'c' e infine 'dir clfonts' elencherà il contenuto delle directory 'c' e 'fonts'.

Titoli in AmigaBasic

Nell'ambiente AmigaBasic è molto difficile cambiare il titolo di una Window una volta che questa è stata aperta e comunque impossibile rinominare il titolo dello schermo. Questo programma usa una routine delle librerie del sistema per aggirare questa mancanza del Basic.

Per il funzionamento del programma dovete avere il file 'intuition.bmap' nella directory 'libs:' del disco Workbench o nella stessa directory nella quale si trova il programma. Per creare questo file dovete usare il programma 'ConvertFD' nel disco Extras (directory 'BasicDemos').

```
LIBRARY "intuition.library"
ws&=WINDOW(7) 'puntatore ad una struttura Window
INPUT "Titolo Window";wt$
INPUT "Titolo Screen";st$
wt$=wt$+CHR$(0) 'per nessun titolo settato
st$=st$+CHR$(0) 'stringa a chr$(0)
CALL SetWindowTitles (ws&,SAD-
```

```
D(wt$),SADD(st$))
'SADD ritorna l'indirizzo della stringa depositata
LIBRARY CLOSE
END
```

Messaggi segreti di Amiga

Quelle che stiamo per illustrarvi potremmo definirle delle semplici curiosità piuttosto che delle utili notizie. Per prima cosa dovete munirvi del disco Workbench e spostare qualsiasi Window che eventualmente rende invisibile la title bar, quindi dovete clickare sulla finestra del Workbench in modo da attivarla. A questo punto sulla title bar compare il solito messaggio 'Workbench release...', con l'ammontare della memoria ancora disponibile. E' proprio qui che compaiono i messaggi segreti dell'Amiga.

Purtroppo per poter visualizzare questi messaggi dovete sviluppare tutte le vostre doti acrobatiche, specialmente per quanto riguarda le dita delle mani. Per prima cosa dovete tenere premuti i tasti SHIFT e ALT (sulla sinistra) contemporaneamente con il pollice della mano sinistra, mentre con il pollice della mano destra dovete tenere premuti simultaneamente i tasti SHIFT e ALT sulla destra. Ora con il dito medio della mano sinistra premete il tasto F1, tutti cinque i tasti devono essere a questo punto premuti. Dopo un flash dello schermo compare il messaggio segreto nella title bar, provate quindi a lasciare F1 e premere F2 e così via per tutti e dieci i tasti funzione, ciascuno di essi contiene una scritta.

Se dopo aver visualizzato questi messaggi avete messo a dura prova l'elasticità delle vostre dita, non disperate, vi sarà servito d'allenamento per ottenere le prossime scritte. Assicuratevi di nuovo che la window del Workbench sia attivata, la title bar visibile e sopratutto che il dischetto del Workbench sia inserito nel drive interno. A questo punto, entra in gioco anche il mouse, il quale deve essere posizionato vicino ai tasti SHIFT e ALT (sulla destra) in modo che il puntatore sullo schermo sia posizionato sul secondo gadget da destra sulla title bar, ovvero quello per lo sfoglio delle schermate. Ora siete pronti per la fase sucessiva, innanzitutto dovete ottenere il messaggio segreto contenuto in F1 seguendo la procedura di cui sopra. Con tutti e cinque i tasti premuti provate con un dito della mano destra ad estrarre il disco Workbench dall'Amiga, ed ecco apparire un'altra scritta segreta sulla title bar. Non muovetevi, rimanete con le dita incollate alla tastiera!

Se siete pudici vi consigliamo di non attuare quanto scritto nelle seguenti righe, se invece decidete di continuare in questa strana esplorazione di Amiga non scandalizzatevi troppo per ciò che vi capiterà di leggere. Sempre mantenendo le dita nella posizione appena illustrata (il puntatore del mouse deve trovarsi sempre sul gadget per lo sfoglio delle schermate), cercate di premere di nuovo il tasto sinistro del mouse e in qualche modo reinserite il disco Workbench nel drive. L'osceno messaggio appare; vi state chiedendo che cosa avranno voluto dire con tale scritta?

Nell'eventualità che vi riesce difficile attuare le suddette istruzioni potrete sempre rivolgervi a qualcuno di passaggio nei paraggi: se per caso non la pensa come voi fategli togliere gli occhi dallo schermo!!

Come non perdere lo schermo Workbench

Sicuramente vi sarà capitato di voler uscire da qualche programma per ritornare all'uso del Workbench e accorgervi che la schermata con le varie icone è scomparsa nel nulla. Di solito questo accade quando si usano dei programmi che occupano molta memoria e che perciò chiudono il Workbench a caricamento ultimato. Naturalmente questa scelta fatta dai vari programmatori di package come De Luxe Music o ABasic, può talvolta dare fastidio. Un metodo infallibile per evitare questo inconveniente è costituito dall'aprire una finestra CLI sullo schermo Workbench e di proseguire nel caricamento del programma "senza ritorno". In questo modo il sistema trovando una finestra aperta non potrà chiudere la schermata Workbench e, pertanto, una volta selezionato il Quit dal programma vi ritroverete di fronte alla window CLI contornata dalle icone dei dischi inseriti nei drive.

WORKSTATION AMIGA MIDI

di Aldo e Andrea Laus

Abbiamo visto nella puntata precedente una panoramica delle molteplici applicazioni MIDI in cui un computer può essere impiegato e la tipologia del software che supporta tali applicazioni.

Entriamo ora nel dettaglio dei componenti hardware per realizzare la vera e propria workstation MIDI.

Il primo accessorio indispensabile ad un computer per interagire con il mondo MIDI è appunto un'interfaccia MIDI.

Cerchiamo di definire il ruolo e le prestazioni di questo componente, senza necessariamente scendere in dettagli troppo tecnici come purtroppo capita spesso quando si tocca questo argomento.

Interfaccia MIDI Cos'è e a cosa serve

I moderni strumenti musicali elettronici (sintetizzatori, pianoforti,drum machines ecc.) sono ormai equipaggiati di serie con il MIDI e, al di là della loro tipologia ed aspetto fisico esterno, si possono considerare costituiti essenzialmente da un computer, molto simile al nostro ben noto Amiga.

Computer sì, ma programmato in modo fisso (mediante ROM) per svolgere una funzione piacevole: produrre musica.

Invece della tastiera QWERTY infatti, anche dal punto di vista hardware troviamo i tasti bianchi e neri e dei tastini di comando, i cui compiti sono prefissati.

Anche la funzione monitor è svolta da un piccolo display LCD finalizzato allo scopo.

In altre parole sia gli input che gli output di questi veri e propri computer musicali sono specializzati nel loro campo applicativo specifico.

Analogamente a quanto avviene nel mondo dei personal computer quando si vuole far comunicare due macchine fra di loro (PC-stampante; PC-PC ecc.), anche nel mondo degli strumenti musicali (leggi computer dedicati musicali) c'è la necessità di realizzare questo dialogo: perciò per questi ultimi è stato definito a livello internazionale uno standard di comunicazione chiamato MIDI (Musical Instrument Digital Interface) che stabilisce sia l'hardware che il linguaggio di comunicazione fra le unità che devono essere collegate.

Gli strumenti musicali elettronici MIDI sono quindi nell'accezione più semplice dei computer dotati di interfaccia verso l'esterno, attraverso la quale, con un opportuno linguaggio possono comunicare con i loro simili.

Infatti, senza la minima difficoltà, basta collegare fra loro per mezzo di un cavetto MIDI due tastiere per verificare che il dialogo reciproco è immediato e che il musicista non è assolutamente tenuto a conoscere l'informatica fino a questo punto.

Quando però il musicista scopre (o per caso al SIM o allo SMAU o presso un collega) cosa si può fare con un computer collegato ai propri strumenti MIDI beh, allora non lo trattiene più nessuno ed inizia la ricerca affannosa (perché i testi sono rari) per saperne di più.

Generalmente, salvo rarissimi casi di computer multipurpose ma con forte vo-

cazione musicale, i computer non nascono compatibili con gli strumenti musicali MIDI dal punto di vista hardware e quindi è necessario interporre fra lo strumento e il computer un circuito adattatore che svolga funzioni di interprete fra le comunicazioni delle due macchine suddette.

Questo circuito è chiamato INTERFACCIA MIDI.

Pensiamo che a questo punto valga la pena di fare una piccola digressione storica per offrire ai neofiti qualche elemento utile a conforto delle loro scelte future.

Chi sa già passi pure oltre, senza offesa perbacco!

L'interfaccia Dr. T's model A inserita nella Workstation MIDI

L'INTERFACCIA Midi

Si può dire che dalla nascita del sistema MIDI (1983) sono comparse via via le diverse interfacce MIDI per i vari computer che mano a mano sono comparsi all'orizzonte ed ancora oggi siamo in fase di evoluzione.

All'inizio c'erano l'Apple II, il Commodore 64 e lo Spectrum e già allora si vide subito un proliferare di interfacce per questi computer, di origine USA, UK, Germania, Giappone e Italia.

Per rimanere in tema con l'indirizzo della nostra rivista ed analizzando più da vicino quindi il filone Commodore, forse il maggior numero di proposte furono rese disponibili per il C64 vista la popolarità di questo computer.

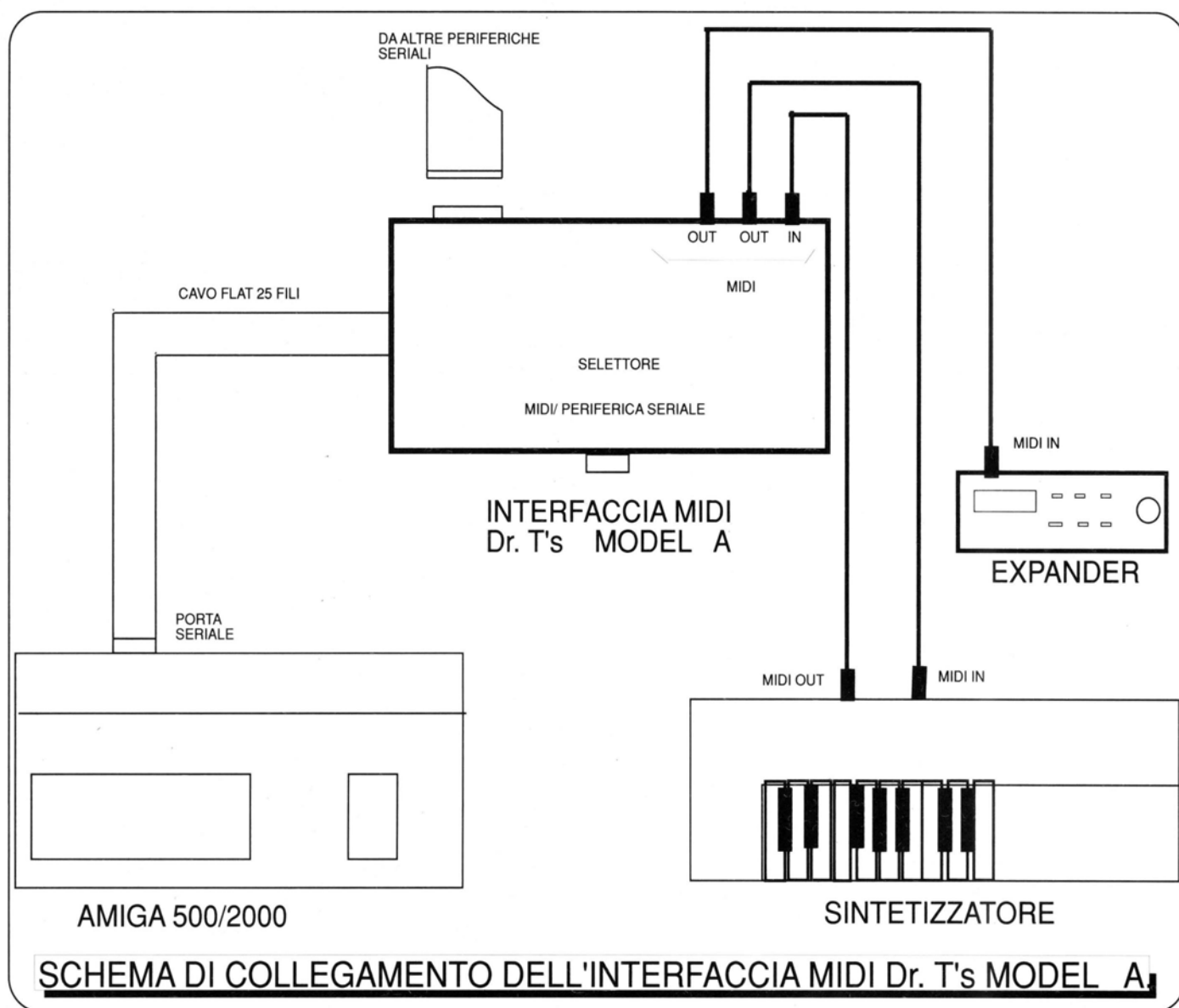
La tipologia dei prodotti era orientata in due direzioni:

1) Interfaccia con hardware ridotto all'essenziale ovvero una presa MIDI IN e una presa MIDI OUT verso il mondo MIDI e attacco alla presa parallela del C64, quella usata comunemente dalle cartucce.

2) L'interfaccia con hardware più sostanzioso che offrisse più flessibilità nei collegamenti verso il mondo MIDI con 1 MIDI IN, fino a 3 MIDI OUT, SINCRONIZZAZIONE FSK ecc., fermo restando il collegamento alla presa parallela del computer.

Vista anteriore dell'interfaccia Model A

Vista posteriore dell'interfaccia Model A

SCHEMA DI COLLEGAMENTO DELL'INTERFACCIA MIDI Dr. T's MODEL A

Purtroppo, motivi sia tecnici che commerciali hanno determinato il proliferare di diversi sistemi di collegamento fra interfaccia e computer per cui non tutti i programmi per il C64 girano con le stesse interfacce.

Questa debolezza fu prontamente sfruttata da un concorrente che, inglobando nel computer l'hardware di interfaccia, mise di fatto tutte le software house di fronte ad un prodotto standard, con grande soddisfazione e beneficio dell' utenza.

Oggi Amiga ha raccolto la sfida e si presenta con un hardware in grado di svolgere quasi tutte le funzioni che nel C64 venivano affidate ad una piastra elettronica fuoribordo o meglio, a nostro avviso, le funzioni più critiche (generazione frequenza clock a 2 Mhz., circuito ACIA per la conversione parallelo/serie e viceversa), lasciando l'onere, comunque lieve, di integrare il computer con un hardware molto semplice solo a coloro che intendano midizzarsi.

In altre parole con Amiga, previo l'utilizzo di opportuno software, è possibile servirsi di due terminali della porta seriale per trasmettere e ricevere i segnali MIDI, salvo la necessità di provvedere esternamente a dei banali circuiti per separare tali terminali elettricamente rispetto al mondo esterno MIDI.

A conferma della validità di questa standardizzazione, le migliori software houses musicali stanno immettendo sul mercato sia nuovi programmi che sfruttano in pieno le funzioni multitasking di Amiga che versioni adattate di programmi gia di successo su altre macchine.

La possibilità di agganciarsi in modo standard e con semplicità circuitale ad Amiga per il MIDI dal lato del computer, ha comunque lasciato la possibilità di proporre interfacce con diversi livelli di prestazioni dal lato MIDI, fra le quali l'utente può scegliere in funzione della configurazione del sistema che intende realizzare.

Ricordiamo infatti che, anche dispo-

nendo di un computer con interfaccia MIDI incorporata, ma con una sola presa IN e OUT, è verosimilmente necessario prima o poi interporre qualche box di espansione fra computer e strumenti per moltiplicare gli accessi quindi, bene o male, lo scatolino in più prima o poi ci scappa!

Nota tecnica sull'interfaccia midi

Il protocollo MIDI, ovvero l'insieme dei messaggi che costituiscono il linguaggio MIDI, viene utilizzato fra gli strumenti musicali interconnessi per mezzo della trasmissione digitale seriale di dati in modo asincrono, alla velocità di 35,250 BAUD (Bit*sec.).

Il linguaggio MIDI è basato su microprocessori a 8 bit e, a seconda del tipo di messaggio da trasmettere, vengono inviati uno, due o tre byte.

Quando uno strumento deve inviare dei messaggi, un circuito denominato ACIA (Asyncronous Complex Interface Adaptor) provvede ad "impacchettare" assieme agli 8 bit utili di ogni byte, altri due bit, uno iniziale settato a zero come bit di start e uno finale settato a uno come bit di stop.

All'altro estremo del cavo, nello strumento ricevente, un circuito analogo ma questa volta presettato in ricezione, dopo averli verificati elimina i due bit di controllo di ciascun byte ricevuto e trasferisce al microprocessore dello strumento musicale ricevente solo gli 8 bit dei dati necessari per decodificare . Se il computer che vogliamo inserire nel sistema non è gia dotato della capacità di eseguire le suddette operazioni sia in ricezione che in trasmissione, è necessario aggiungere nel sistema un dispositivo hardware che chiamiamo: INTERFACCIA MIDI.

Questa da una parte sarà collegata alla porta parallela del computer in modo da:

1- ricevere o trasmettere verso di esso i dati a 8 bit

2- consentire al microprocessore di accedere ai suoi registri interni tramite opportuni indirizzi.

Dall'altra parte si presenterà in linea con il mondo MIDI con una o più prese MIDI OUT e una presa MIDI IN.

Se invece il computer è più evoluto e svolge al suo interno le funzioni viste sopra, tipiche del chip ACIA, allora si può utilizzare la porta di uscita seriale del computer.

In questo caso il confine verso l'esterno si sposta drasticamente in favore della semplicità circuitale dell'interfaccia.

E' questo il caso del nostro computer Amiga.

Per quanto riguarda l'alimentazione dei propri circuiti, generalmente le interfacce prelevano la tensione dal computer a cui sono collegate, tramite opportuni piedini del connettore di interconnessione.

Dr.T Model-A Interfaccia MIDI per i computer Commodore Amiga 500 e 2000

In questo articolo prendiamo in esame una delle prime interfacce MIDI per il vostro computer Amiga, sia 500 che 2000, disponibili in Italia, costruita dalla nota casa americana Dr.T's Music Software,Inc. e messaci cortesemente a disposizione per questa prova dal distributore esclusivo per l'Italia: Music Sound Technology.

L'interfaccia si presenta in un elegante cofanetto di plastica di dimensioni contenute, come potete rilevare dalle foto.

I due collegamenti con l'esterno, lato computer e lato MIDI, sono entrambi su un lato del mobiletto e sono costituiti da:

- un connettore DB 25 volante collegato all'interno dell'interfaccia mediante un cavo flat a 25 conduttori che va inserito nella porta seriale di Amiga
- tre prese MIDI fissate sull'interfaccia e rispettivamente un IN e due OUT alle quali vanno collegati gli strumenti MIDI.
- un connettore D 25 femmina fissato sull'interfaccia, che consente di collegare altre periferiche seriali al sistema in alternativa all'interfaccia MIDI, senza scollegarla ogni volta dall'Amiga.

Sul lato opposto dell'interfaccia c'è un pulsante che consente di selezionare o l'interfaccia oppure la periferica collegata al connettore femmina presente sull'interfaccia.

Questa caratteristica consente una versatilità maggiore rispetto ad altri prodotti in quanto evita la scomodità di scollegare l'interfaccia quando occorre lavorare con altre periferiche.

Di fatto l'oggetto interfaccia entra a far parte del sistema e lo lasciate sempre lì commutandolo, quando serve, col bottone.

Così sono eliminati potenzialmente i problemi che possono derivare dal continuo attacca e stacca !

Potrà sembrare una delusione per chi non ancora addetto ai lavori possa aver pensato il contrario, ma l'unica cosa da fare con un'interfaccia MIDI è giusto il collegamento come descritto sopra e poi basta.

Funziona subito e poi non c'è più nulla da fare.

Abbiamo provato, attraverso l'interfaccia Dr.T's Model-A collegata a diversi tipi di sintetizzatori, tre programmi MIDI su Amiga 500 e precisamente:

- De Luxe Music Construction Set
- Music Recording Studio
- MT 32 Caged Artist Editor

e tutti hanno funzionato alla perfezione; inoltre anche programmi non MIDI hanno dimostrato di funzionare benissimo lasciando l'interfaccia MIDI collegata.

Per i curiosi sul funzionamento tecnico, diciamo che il foglio di istruzioni fornito con l'apparato dichiara che l'interfaccia Dr.T's Model-A funziona con tutti i software che utilizzano la porta seriale Amiga come input e output per i dati MIDI.

La presenza di due prese MIDI OUT in parallelo consente di collegare contemporaneamente due periferiche MIDI, per esempio due sintetizzatori o un piano e un expander, senza far subire ai dati il ritardo dovuto ad eventuali ripetizioni da parte di prese MIDI THRU.

Qualora si debba ampliare il sistema, è consigliabile inserirvi un box patch bay, che risulta particolarmente utile quando si debbano commutare sia molte prese OUT ma anche diverse prese IN.

La realizzazione meccanica dell'interfaccia in oggetto si presenta in linea con lo stile del computer ed i connettori multipin assicurano perfettamente i contatti elettrici evitando grattacapi.

Si giustifica così in pieno quanto asserisce il foglio di istruzioni che, in caso di malfunzionamento del sistema invita, come prima cosa, a verificare i propri cavi MIDI perché di solito il punto debole è fuori!

Il prezzo di listino (Sett.88) I.V.A. esclusa èdi Lire 175.000 che, in un certo senso vanno viste come una gabella indispensabile per entrare nel mondo MIDI.

Per ulteriori informazioni potrete rivolgervi direttamente alla MUSIC SOUND TECHNOLOGY S.r.l. Via 1o Maggio 5, Cavenago Brianza (MI). Tel. 02-95018031.

di Vito Lorusso

Nelle ultime fiere mondiali degli strumenti musicali e dei computer, la COMMODORE era presente con gli stand per le applicazioni musicali dei computer AMIGA 500/1000/2000. L'AMIGA è una macchina tecnicamente valida per le applicazioni musicali sopratutto per l'elevata velocità di gestione delle informazioni relative al codice M.I.D.I. (Musical Instrument Digital Interface) e il Multitasking. Inoltre,le prestazioni ottenibili nelle applicazioni videografiche hanno colpito la immaginazione degli sviluppatori di software; si parla già di programmi in grado di sviluppare applicazioni musicali e videografiche contemporaneamente. Per questo, usando i software applicativi adeguati, il futuro di AMIGA è quello del la trasformazione in stazione di lavoro audio/video, ruolo che finora è stato penalizzato dalla mancanza di software adeguato. Nei prossimi mesi assisteremo alla presentazione di programmi(di cui parleremo) via via sempre più sorprendenti che permetteranno di colmare e in molti casi superare il "gap" rispetto ad altre macchine.

Questo articolo, è l'inizio di una rubrica rivolta alle applicazioni musicali di AMIGA che partirà da alcune indicazioni di carattere generale sull'uso dei "Personal Computer" nella musica e sul codice digitale che ha reso possibile lo sviluppo verticale di queste applicazioni denominato "MIDI". La rubrica si articolerà anche attraverso indicazioni emerse da una partecipazione interattiva dei lettori interessati alle applicazioni musicali e audio/video che intendono operare con un "sistema musicale MIDI" basato sui computer AMIGA 500/1000/ 2000 concettualmente valido per ambiti "amatoriali" e "professionali" nello stesso tempo. Su questo numero , parleremo delle performance presentate al SIM , fiera orientata al settore degli strumenti musicali e allo SMAU, fiera dell' Informatica che si sono svolte tra Settembre e Ottobre a MILANO.

Le performance

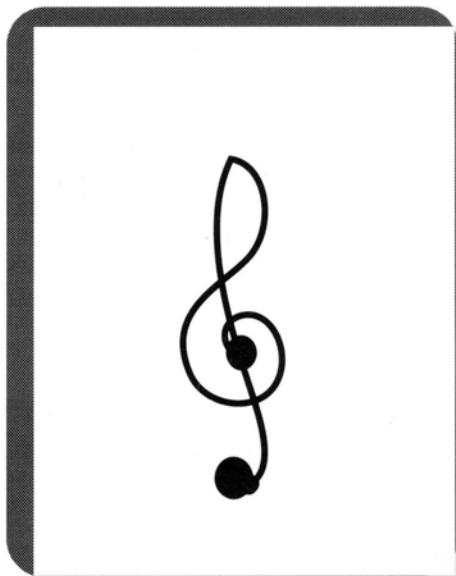
Le due performance sono state diverse sotto l'aspetto tecnico; quella proposta al SIM, consisteva in esecuzioni automatiche di brani che mettevano in luce le capacità di AMIGA come registratore digita-

Le performance musicali di Amiga dal SIM allo SMAU

le MIDI,mentre allo SMAU, la macchina supportava con parti preprogrammate (o programmate dal vivo) e con la gestione dei suoni le esecuzioni di tre musicisti dotati di Controller MIDI: MIDI DRUM, MIDI SAX, MIDI KEYBOARD. Al SIM operava una Workstation MIDI costituita da un computer AMIGA 2000 dotato di interfaccia MIDI e da alcuni expander (unità MIDI in grado di generare suoni sfruttando vari modelli di sintesi) e due campionatori (unità digi talizzatrici audio).Grazie alle possibilità Multitasking di AMIGA è stato possibile accedere al sistema a vari livelli, sfruttando alcuni software per l'editing dei parametri degli expander contemporaneamente al K.C.S. (Keyboard Controlled Sequencer),un programma che permette di simulare con l'AMIGA un registratore digitale MIDI a 48 tracce (il termine re gistrazione in questo caso indica la registrazione di eventi MIDI, non di segnali audio). Tramite l'interfaccia MIDI, che nell' AMIGA consiste in un hardware minimo, il computer era collegato a una tastiera musicale per l'introduzione dei dati e a tutte le altre unità del sistema tramite un MIDI THRU BOX, un dispositivo che permette di limitare i ritardi relativi alla trasmissione seriale delle informazioni,il si stema di trasmissione previsto dal codice MIDI (la trasmissione seriale delle informazioni MIDI è vantaggiosa perché consente collegamenti semplici e di conseguenza a basso costo, presenta però alcuni svantaggi nella velocità di tra sferimento delle informazioni).

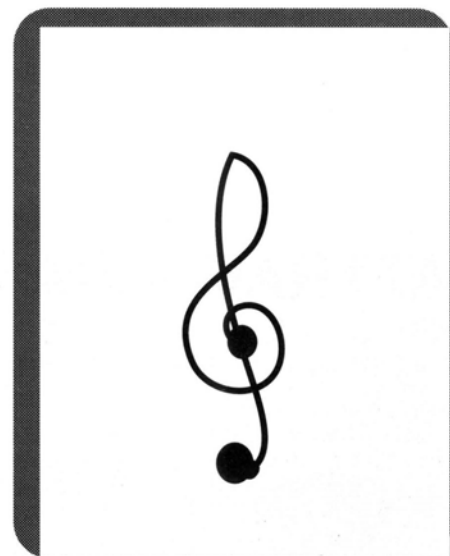
La Performance al SIM

Nella performance al SIM, AMIGA "eseguiva" automaticamente alcuni brani che sono stati registrati nel computer utilizzando la tecnica di registrazione multitraccia del "KCS": un software musicale che sfrutta come tanti altri programmi, la

similitudine operativa con un registratore multitraccia, infatti i comandi fondamentali sono uguali ai "trasporti" di un registratore audio convenzionale: Play, Stop, Pausa, Rec. Nella programmazione dei brani, sono state usate le due possibilità operative di Input dei dati MIDI: RealTime e Step by Step. In RealTime, il musicista ha la possibilità di registrare nell' AMIGA tutti gli eventi MIDI relativi alla sua esecuzione durante lo svolgimento di quest'ultima, mentre in "Step", la programmazione avviene in tempo differito. Durante l'esecuzione automatica dei brani, il computer sfruttando i messaggi MIDI relativi al"Program Change", inviava a tutte le unità del sistema (anche a un processore audio per la creazione di effetti particolari come il riverbero ecc.) i messaggi relativi alle configurazioni e ai suoni da usare nei vari punti dei brani.

La Performance allo SMAU

Durante lo SMAU, è stata proposta una dimostrazione "live" delle applicazioni musicali di AMIGA in un padiglione già strutturato all' insegna della futuribilità e spettacolarità delle macchine con uno stand in cui tre musicisti del "Sound Movement Group" si sono avvalsi di linguaggi multimediali. Il "MIDICONCERT", titolo della performance, tenuto da Vito Lorusso (MIDI Keyboard), Angelo Losasso (MIDI Drum) e Angelo Ricciardi (MIDI Sax), è stato supportato da elementi scenografici quali luci sequenzializzate e una singolare installazione video costituita da 6 videomonitor posizionati su un "totem" che proiettavano immagini del concerto elaborate con programmi videografici di AMIGA. Il sistema musicale MIDI era sostanzialmente simile a quello usato al SIM; in più c'erano alcuni Controller che tramite AMIGA pilotavano le unità MIDI per la generazione dei suoni: -Midi Drum è in pratica una batteria convenzionale che può tramite un'interfaccia appropriata associare un messaggio "Key On" del codice MIDI(messaggio che indica l'attivazione di una o più note)ad ogni superfice battente; -Midi Sax è un Controller simile a un Sax Contralto tradizionale che può trasmettere alcuni messaggi MIDI come il Key On; -il terzo Controller consisteva in una normale tastiera MIDI. Nel Midi Concert, i musicisti, tramite i Controller potevano pilotare le unità MIDI con le configurazioni predisposte da AMIGA. Il computer eseguiva anche alcune parti preprogrammate e in alcuni casi con l'ausilio di funzioni per l'editing in RealTime le parti potevano essere modificate o programmate dal vivo.

Prossimamente

Il discorso AMIGA: Applicazioni Musicali Professionali è destinato a essere sviluppato a livello mondiale;le software "House" specializzate nelle applicazioni musicali hanno già annunciato la imminente commercializzazione di programmi dotati di caratteristiche eccezionali come il " MUSIC X ", un software musicale che sfrutta l'analogia con un registratore multitraccia, consente la manipolazione di 250 Sequenze in grado di contenere informazioni relative ai 16 canali MIDI, è sincronizzabile con il codice temporale per applicazioni audio/video SMPTE (può essere letto direttamente dall'AMIGA),contiene una programma di "libreria" suoni universale e permette di costruire programmi di "Editor" per qualsiasi tastiera o expander MIDI. In più, per lo sviluppo delle applicazioni audio/video, è prevista la sincronizzazione con un software applicativo che consente di creare animazioni in 2 e 3 dimensioni. In ITALIA,il discorso applicazioni musica li continuerà con un laboratorio creato dall'autore di questa rubrica e delle dimostrazioni al SIM e allo SMAU di cui parleremo sul prossimo numero.

SENZA RETE

SENZA RETE

SENZA RETE

ATTRAVERSANDO L'UNIVERSO MUSICALE SUL FILO DELL' INFORMATICA:

Mauro Graziani

di Mr. LAMBDA

Dopo aver presentato software, algoritmi, hardware, linguaggi che in qualche modo erano in relazione all'universo musicale, ci siamo voluti inoltrare nel vivo del confronto tra computer e musica presentandovi un incontro con un artista che consideriamo emblematico proprio per il vivere in prima persona e da protagonista questo confronto.

E' proprio in un' occasione che lo vedeva ancora una volta interprete/creatore dei processi osmotici in atto tra i due universi - questa nostra intervista, infatti, è stata effettuata in occasione di un concerto di musica contemporanea, introdotto da Giuseppe Di Giugno, tenuto al Teatro Comunale di Monfalcone il 26 febbraio 1988, con la partecipazione di Mauro Graziani, Roberto Fabbriciani, Alvise Vidolin - che abbiamo voluto affrontare alcuni problemi relativi al mondo musicale informatico: si va da qualche consiglio a coloro che intendono diventare musicisti informatici ai linguaggi per la programmazione del suono, da alcune considerazioni sulla situazione dell'informatica in Italia alle diverse concezioni di sintesi sonora, dall' importanza dell'informatica per la musica contemporanea alla poetica dell' artista. Ma prima di imboccare la trama delle sue considerazioni vogliamo complimentarci con Mauro Graziani per la sua capacità di tenere le redini dei due mondi, informatico e musicale, e per la sensazione, prodotta dalle sue performance, di essere alle soglie di un mondo nuovo.

A.M. *Come e quando è nato questo suo interesse specifico per la computer music.*

M.G. Io ad un certo punto sapevo fare soltanto musica come professione. Ho studiato violino da piccolo e sono arrivato quasi fino all'ottavo anno che avevo circa sedici anni. A questo punto della mia vita ho incominciato ad interessarmi di musica elettronica.

Sono finito in un posto dove costruivano un tipo di sintetizzatori. Ho lavorato in quel posto qualche mese e me ne sono portato a casa uno. Poi sono piombato a Padova per caso nel '76. In quell' anno c'erano alcune persone dell'università che avevano incominciato a studiare la simulazione della voce con il computer per uso industriale. E da lì si è partiti con l'idea di provare ad utilizzare il computer per la gestione del suono e quindi anche per la produzione della musica. Il computer era un mainframe IBM 370/158. Su questo mainframe abbiamo fatto girare i programmi di sintesi sonora che esistevano in quel periodo: Music V e Music 360. Entrambi erano prodotti americani. Per esempio Music V era stato sviluppato alla Bell Telephone. A questo punto inizia per me tutta l'avventura della computer music: facendo routine in Assembler per montare Music V sull' IBM 370.

A.M. *Tutte queste conoscenze di informatica che man mano ha acquisito hanno avuto un incidenza sul suo stile musicale e nella concezione del brano musicale.*

M.G. Sì, completamente.

A.M. *Quindi c'è stata un'interazione continua tra progressive acquisizioni di informatica e poetica musicale.*

M.G. La situazione in realtà è questa: non so se mi piace di più la musica o il computer.

E ciò infastidisce i musicisti. Sono sicuro che, se per qualche strana ragione mi capitasse o fossi costretto a non fare più musica, lavorerei lo stesso con il computer. Però non sono altrettanto sicuro, se non potessi più lavorare con il computer, di fare ancora musica.

A.M. *Si può parlare di identificazione dei due mondi.*

M.G. Sì, il bello di fare musica con questi sistemi è praticamente quello che è il bello del computer; cioè che tu ti puoi inventare delle cose e realizzarle almeno a livello di simulazione.

Quindi io posso fare delle ipotesi e poi posso lavorare sul suono andando a manipolare persino al livello del singolo timbro. Quindi posso fare sia ipotesi formali generali del tipo costruiamo una struttura di un certo genere, sia delle ipotesi molto più dettagliate che coinvolgano i singoli suoni.

A.M. *Ecco, a questo proposito, in che rapporto stanno le emergenti nuove tecniche di tipo informatico con le precedenti tecniche materiche, elettroniche, gestuali?*

M.G. Rispetto alle vecchie tecniche ci sono più possibilità e ci dovrebbe essere una maggiore coscienza di quello che si fa.

L'analogico è fortemente del tipo prova e vedi quello che viene fuori. Lavorando con il computer si è costretti invece ad avere una coscienza maggiore dell'obiettivo che si sta cercando di raggiungere perché bisogna fare un programma o utilizzare un programma in modo tale da raggiungere questo obiettivo.

E molto più difficile servirsi del ' tenta e riprova ' finchè non si è raggiunto il risultato voluto sia perché questo richiede del tempo e la programmazione stessa richiede del tempo, sia perché le possibilità che ci mette a disposizione il computer sono molto più vaste di quelle offerte dai mezzi analogici.

Con l'elaborazione analogica si possono avere a disposizione per esempio quattro o cinque oscillatori, a seconda del sintetizzatore utilizzato, oppure, nei grossi studi se ne possono utilizzare fino a trentuno. Digitalmente se ne hanno invece a disposizione un numero infinito.

Chiaramente bisogna fare una distinzione preliminare tra tempo reale e tempo differito.

Se si vuole utilizzare il computer per la produzione e manipolazione del suono in tempo reale, i limiti si ripropongono, anche se non nelle stesse proporzioni e con le medesime caratteristiche. Se si è disposti, invece, ad aspettare, e quindi ad utilizzare programmi che ricevano dati, cioè la propria partitura, calcolino tutti i campioni sonori, li mettano per esempio su un hard disk, e quindi vengano ascoltati, il che vuol dire un intervallo che può variare da alcuni minuti a delle ore, fino anche a otto o dieci ore in relazione alla complessità e lunghezza di quello che viene calcolato tra la tua stesura della partitura e l'ascolto del risultato, se si è disposti ad accettare questo, allora si ha semplicemente tutto a disposizione. Nel senso che si può ipotizzare anche l'utilizzo di diecimila oscillatori, basta aspettare il tempo perché il computer si calcoli tutti i campioni di diecimila oscillatori.

Con il tempo differito in pratica non c'è un limite. Per cui è molto difficile se non impossibile utilizzare il ' tenta e riprova '.

A.M. *E chiaro quindi che questa mag-*

giore flessibilità e potenza fa preferire l'approccio informatico e quindi l'abbandono delle vecchie tecniche.

M.G. Per me è preferibile.

A.M. *A questo punto, dato che si è già inquadrato sufficientemente il problema, cerchiamo di chiarire quali siano le conoscenze che secondo lei sono indispensabili a un musicista che intenda affrontare questa materia o quest'area.*

M.G. Premetto che stiamo cercando di fare dei programmi per mezzo dei quali al musicista siano limitate al minimo indispensabile le conoscenze informatiche richieste.

Questo è ciò verso cui l'industria sta andando.

Per esempio, l' Atari adesso mi chiede un sistema per programmare i timbri dei sintonizzatori in FM tipo Yamaka senza vedere un numero.

Cioè avendo a disposizione una serie di cursori uno dice: voglio un attacco più o meno percussivo, o cose del genere. Questa prospettiva può andare benissimo per il punto di vista del musicista che si considera ' musicista ' e che utilizza queste macchina per determinare certi mondi sonori controllabili con il minore dispendio di energie possibile. Però io sono uno di quelli che crede che per fare informatica musicale ad un livello serio bisogna saper programmare, e bene. Tanto.

A.M. *Ci sono dei linguaggi che ritiene più adeguati a questo tipo di realizzazioni, oppure l'approccio linguistico scelto è indifferente.*

M.G. A chi parte adesso dico di incominciare con il Lisp, perché stiamo portando sui Personal i sistemi in Lisp, che sono molto più interessanti di quelli che ci sono adesso.

Oppure alcuni potenti derivati di questo linguaggio, come il Logo, che nonostante sia considerato un linguaggio da bambini, in realtà non lo è.

A.M. *E il C, con il suo grande mito della portabilità ...*

M.G. Il C è il linguaggio di adesso.

A.M. Ma il futuro lo vedo in Lisp.

M.G. Il futuro lo vedo nei linguaggi altamente simbolici, e non dico nei linguaggi non procedurali come il Prolog, perché non si sa ancora bene che cosa si potrà ottenere musicalmente dal Prolog. Nel senso che si stanno ancora facendo esperimenti per utilizzare il Prolog musicalmente. Il problema è che non essendo procedurale può per ora essere utilizzato preferibilmente in fase compositiva, come una immensa gestione di informazione del tipo: lo dico che il mio materiale è fatto così e così, io dico che fra le varie note e sezioni dei pezzi devono intercorrere queste relazioni. Il linguaggio è in grado di fornirmi quali possibilità mi possono risultare dalla connessione tra regole e materiale dato. Questo è tipicamente Prolog.

A.M *Ma come si comporta per esempio nel determinare la forma d'onda?*

M.G. Ci sono delle difficoltà ogni volta che si affrontano problemi fortemente matematici.

A.M. *Oggi come oggi, quale è il suo linguaggio preferito?*

M.G. Io utilizzo il C, anche quando devo fare dei programmi commerciali. Ho fatto delle cose in Forth perché mi hanno fornito un sistema multitasking e mi sto interessando al Modula-2 che ritengo interessante. Ho fatto pure delle cose in Lisp e il Lisp mi piace.

A.M. *Passiamo ora dai linguaggi alle tecniche di sintesi del suono: qual è secondo lei la tecnica che realizza meglio la sintesi del suono?*

M.G. Questo è un po' difficile da dire perché bisogna vedere dove si vuole arrivare. Dunque: l' FM è stata sviluppata così pesantemente che credo che adesso sia la cosa su cui ci sono più conoscenze e credo che essa sia la tecnica di sintesi, non di campionamento, più potente che abbiamo a disposizione. Anche perché i sintetizzatori in FM continuano ad essere costruiti molto meglio degli altri. Poi c'è la faccenda del campionamento ...

A.M. *Ed in questo modo andiamo a trattare della sintesi additiva.*

M.G. Già, in questo campo è importante che sia possibile campionare a 44.1 MHz. Non è importante andare oltre tale soglia dal momento che udiamo fino ai 16, 20 MHz, e poichè tale frequenza di campionamento ci permette una banda fino ai 22.5 MHz possiamo ritenerci soddisfatti.

A.M. *E il problema delle armoniche ...*

M.G. Dopo i 22 MHz non le sentiamo.

A.M. *Ma l'influenza non potrebbe essere infra-uditiva? Cioè che la loro influenza si faccia sentire sulla globalità del suono?*

M.G. E' proprio ciò che ci si sta chiedendo. Io personalmente sono piuttosto scettico al proposito. E anche una questione fisiologica. Il punto base è questo: bisogna vedere se siamo influenzati da aspetti fisiologici o psicologici. Se il limite è fisiologico, allora non c'è nulla da fare.

A.M. *La sintesi additiva si sostituirà alla modulazione di frequenza di Browning?*

M.G. La sintesi additiva ' vera ' non è stata ancora realizzata in un sintetizzatore commerciale. La sintesi additiva ' vera ' consiste nell'avere a disposizione per ogni armonica un oscillatore con il suo inviluppo sia in ampiezza che in frequenza, il che significa che certe note basse del pianoforte possono avere anche settanta parziali. Ora per fare queste note del pianoforte con la sintesi additiva classica abbiamo bisogno di 70 oscillatori e 140 inviluppi e una potenza di calcolo equivalente. A suo tempo abbiamo configurato il Music V con circa 1500 oscillatori simultanei, naturalmente non in tempo reale, ma in tempo differito, perché abbiamo calcolato che questo è il numero di componenti necessario a rendere una complessità sonora pari a quella di un' orchestra. Ora si può capire perché questo tipo di sintesi additiva non sia stato ancora realizzato commercialmente. Ci sono delle altre tecniche che forse sostituiranno anche l'FM.

A.M. *Facciamo una constatazione preliminare: la diffusione dell'informatica per mezzo di home e personal computer ha allargato di conseguenza anche la cer-*

chia delle persone che sono interessate, sia da fruitori che da creatori, alla musica elettronica e alla computer music. Può essere questo fatto la causa di un eventuale maggior interesse verso l'area sperimentale della musica. E come guarda il concertista tradizionale questo fenomeno?

M.G. Io sarò molto cattivo, ma ho l'impressione che l'interesse sia aumentato perché il computer è sulla cresta dell'onda, e spero che ci rimanga per almeno un altro po' di anni, almeno fino a quando mi divertirò ancora ad esibirmi e a tenere concerti. Secondo me il pubblico non è mai stato numeroso nei concerti di musica sperimentale e penso che le cose non siano mutate di molto. Se si rileva una maggiore presenza questa è legata al fattore curiosità e novità relativa all'utilizzo del computer. Tutto ciò, però, richiede naturalmente una conferma e un consolidamento di tendenza che verificheremo nel tempo.

A.M. *Il divario tra la computer music accademica, o diciamo istituzionale, e quella commerciale rimane invalicabile oppure è in atto un qualche processo osmotico, una compenetrazione cioè che permetta il costituirsi di aree sperimentali nuove e che nello stesso tempo sappia tener conto delle diverse esperienze.*

M.G. Che sia in atto una compenetrazione è innegabile. Però non siamo ancora in grado di dire verso dove vada e che portata abbia. Non tutto ciò che viene prodotto in quest'area è però degno di considerazione.

A.M. *In questo momento è possibile ed ha un senso autenticamente artistico e non solo tecnologico un'autonomizzazione del discorso musicale informatico.*

M.G. Esiste la possibilità di incominciare a farlo.

A.M. *A quali difficoltà va incontro tale orientamento musicale, tenendo presente il fatto che autori come Berio e Nono, tanto per citarne alcuni, hanno preferito e continuano a preferire tecniche miste in cui il computer o la macchina è sempre affiancata dagli strumenti tradizionali.*

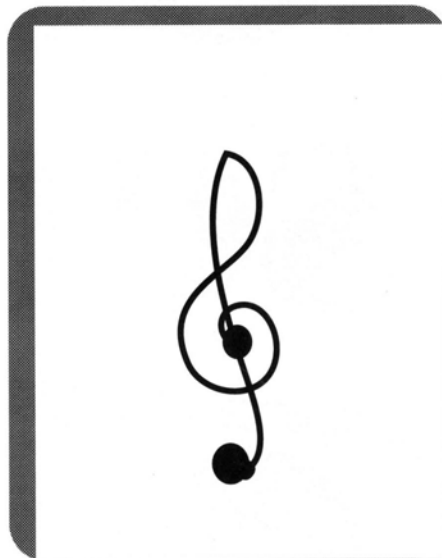
M.G. Ho lavorato per composizioni di Donatoni, Clementi e Nono. Il fatto di utilizzare l'informatica come se fosse uno strumento qualsiasi è abbastanza evidente in due casi: nel momento in cui il musicista pensa al risultato musicale e non ai modi in cui lo ottiene oppure quando il musicista ha ormai raggiunto un'esperienza compositiva a cui non è disposto a rinunciare per intraprendere nuove strade.

A.M. *Quale è lo spazio riservato alla formazione orientata a queste nuove acquisizioni, nei conservatori per esempio ...*

M.G. La situazione varia da conservatorio a conservatorio. In Italia ci sono una decina di cattedre di musica elettronica. Di queste dieci cattedre solamente il cinquanta per cento è dotato di attrezzature adeguate, ma gran parte di questa dotazione è composta da strumenti di tipo commerciale, certamente validi da un punto di vista sonoro, ma scarsamente indicati per la ricerca pura. Per la ricerca abbiamo bisogno di una macchina configurabile, per mezzo della quale si possa fare delle ipotesi. Non c'è nessuna cattedra dotata attualmente di uno strumento di questo tipo. Allo stato attuale la situazione è grave.

E non cambierà fintantochè i conservatori non saranno smantellati e verranno creati dei dipartimenti di musica presso le univerità. Perché è proprio nelle università che ci sono gli strumenti per fare ricerca.

A.M. *Una persona che ha fatto l'esperienza dello spazio sonoro e dell'universo musicale attraverso una manipolazione diretta ed arbitraria, come guarda poi la struttura armonica e melodica tradizionale?*

M.G. Si impara ad ascoltare le cose più in senso timbrico. E questo alla lunga influenza in modo positivo anche il modo di concepire l'architettura e le relazioni dei brani musicali.

A.M. *Come si potrebbe caratterizzare la sua più recente attività?*

M.G. Io parto dall'idea che, dato che abbiamo un mezzo che può gestire sia la parte sonora che la parte grafica, allora è interessante considerare l'idea di metterle insieme sotto il controllo di uno stesso programma, facendo in modo che il risultato possa essere interpretato per generare parti sia grafiche che sonore. In pratica io mi invento delle associazioni, completamente arbitrarie; anche se inevitabilmente tengo implicitamente conto della mia esperienza. Anche perché non esistono delle associazioni precostituite tra grafica e suono, e quindi il compito è quello di produrre delle associazioni che possano essere significative, e che costituiscano, insieme, un mondo grafico-sonoro che abbia un senso percettivo, cioè che permetta di individuare le relazioni che lo sottendono. Quindi questo mondo di relazioni deve permettere di individuare le connessioni tra eventi sonori e grafici, poi spetta alla mente dello spettatore fare il resto.

Scheda biografica.

Graziani Mauro è nato a Verona il 24/1/1954 ed effettua i suoi studi musicali al Conservatorio B. Marcello di Venezia con Alvise Vidolin.

Dal 1970 lavora nell'area della musica elettronica e dal 1976 opera presso il centro di Sonologia Computazionale dell' Università di Padova come compositore e ricercatore.

Finora ha ultimato varie ricerche finalizzate all'impiego delle tecnologie avanzate in campo musicale e composto opere, sia di Computer Music che di tipo multimediale, eseguite e radio-diffuse in Italia ed all' estero (Europa, America, Est europeo).

Ha ricevuto commissioni da parte del LIMB-Biennale di Venezia (1980 - composizione "The silent God") e dalla RAI (1982 - composizione "Trasparenza").

E' stato selezionato per la partecipazione alla prima edizione della manifestazione "Venezia Opera Prima".

Ha ottenuto il primo premio al Concorso Nazionale di Musica Elettronica "Città di Abbadia S.Salvatore 1985" con la composizione "Wires".

Le sue opere "Winter Leaves", "The Silent God" e "Landing" hanno ottenuto menzioni al nono e undicesimo International Electroacoustic Music Awards di Bourges.

In qualità di esecutore tecnico all'elaboratore ha realizzato l'opera "Parafrasi" e la parte su nastro di "Fantasia su roBErto fABriCiAni", entrambe composte da Aldo Clementi, "Canzone Veneziana" di Joel Chadabe e "Elettronico" di Franco Donatoni (parte dell'opera "Atem").

Con A. Vidolin e S. Sapir, ha inoltre collaborato alla realizzazione della parte di musica informatica ed alla messa in scena delle due versioni finora allestite del "Prometeo" di Luigi Nono.

Ha pubblicato articoli su varie riviste del settore e tenuto conferenze in varie sedi.

Attualmente, svolge attività compositiva e di ricerca presso il Centro di Sonologia Computazionale dell' Università di Padova e nel suo studio a Verona.

Musiche di Mauro Graziani sono disponibili su disco EDIPAN PRCS20-16 (serie INSOUND).

Gioca sul serio!

D.O.C.

Se vuoi essere sempre super-informato e super-aggiornato,
se vuoi divertirti e giocare, fallo sul serio.
Cerca sempre i prodotti Jackson Software D.O.C. perché solo
Jackson ti garantisce serietà e giochi originalissimi, esperienza
e novità, informazione e divertimento. E poi, diffida delle cattive
imitazioni, delle cassette pirata e delle edizioni fantasma.
Se non vuoi brutte sorprese non scegliere a caso,
non sprecare tempo e denaro. Chiedi sempre il marchio Jackson
Software D.O.C. Scegli la sicurezza
della qualità che è solo Jackson.

GRUPPO EDITORIALE JACKSON
AREA CONSUMER

Scegli il meglio: scegli Jackson.

CONVERSIONE IFF ICONA-SPRITE

Ecco a voi IFFConverter una utility per trasformare i brush in altri e più utili oggetti

di Paolo Russo

Ecco due domandine facili facili le cui risposte, guarda caso, coincidono: qual'è il più potente e pratico programma di grafica esistente per l'Amiga?

E qual'è, in assoluto, il più famoso programma tra la miriade di quelli disponibili per il medesimo computer? Risposta pressochè unanime: "Deluxe Paint II!".

Grazie ai canali ufficiali di importazione e soprattutto a quelli non ufficiali, non esiste praticamente utente Amiga che non possieda una copia di questo famoso programma, che agevola enormemente la creazione di ogni tipo di immagine.

Appare quindi paradossale che per disegnare un'icona sia necessario impazzire con il rozzo IconEd incluso nel disco dello Workbench, per non parlare di chi fa uso degli sprite in Basic e si trova costretto a disegnarli con l'impraticabile Object Editor reperibile negli Extras.

L'utility presentata in queste pagine consente di utilizzare il mitico Deluxe Paint o altri programmi simili per la creazione di icone e sprite.

La compatibilità IFF ILBM

Chiariamo subito un punto: IFFConverter non è dedicato ai possessori di Deluxe Paint. Ogni genere di tool grafico può essere usato al posto del Mitico appena citato, a condizione che sia con esso compatibile a livello di file. Esiste per costoro un formato standard, chiamato IFF (Interchange File Format), ideato qualche anno addietro dalla Electronic Arts, che dovrebbe risolvere i problemi di incompatibilità potenzialmente presenti tra prodotti

diversi.

Esiste un formato per i testi (IFF FTXT, Formatted Text), per la musica (IFF SMUS, Simple Musical Score), per i suoni campionati (IFF 8SVX, 8-bit Sampled Voice) e ne esistono, ahinoi, due per la grafica: IFF ILBM (Interleaved Bitmap) e IFF ACBM (Amiga Contiguous Bitmap); il primo è utilizzato da Deluxe Paint, il secondo da Graphicraft. In un mondo di tool grafici che risultano inutilizzabili in assenza della completa compatibilità reciproca, era naturale che due diversi standard non potessero convivere.

Un bel giorno uno dei due disse all'altro: "Ehi, amico, questo posto è davvero troppo piccolo per tutti e due" e gli spettatori attoniti li videro dirigersi verso un luogo deserto. Un lungo silenzio, seguito da alcune improvvise detonazioni di arma da pixel; dopo un po' si vide tornare il solo ILBM, stringendo in pugno il suo fido Deluxe Paint con la canna ancora fumante. "ACBM era più agile e veloce", ammise in seguito, "ma la sua arma non poteva competere con la mia". La legge della giungla informatica, talvolta, è assai spietata e ILBM, grazie soprattutto all'enorme numero di sostenitori che decise di mettersi dalla sua parte per timore della sua mitica arma, non fu mai processato, divenne ricchissimo e dette vita ad una pregevole industria.

IFFConverter giunse nel Mondo della Grafica quando il duello era ormai solo uno sbiadito ricordo. Essendo un artigiano e mercante di professione (comprava file IFF e, dopo alcune sostanziali modifiche, li rivendeva nel Paese delle Icone e nel Potentato degli Sprite, Bob e Affini, dove sarebbero stati pagati a peso d'oro) si rese conto che l'unica materia prima disponibile era di marca ILBM, e non si curò neppure dello scarso mercato rappresentato dai pochi eredi dello sconfitto.

Modalità di utilizzo

IFFConverter è scritto in AmigaBasic, in modo che chiunque possa esaminarlo o modificarlo a piacimento. Dopo il run appare una schermata che descrive le opzioni disponibili da menu. Esse sono essenzialmente quattro: caricamento di un brush di tipo IFF ILBM, conversione in formato icona, conversione in formato sprite, estrazione della paletta. Nel seguito si parlerà costantemente di Deluxe Paint, ma ricordate che il discorso vale per qualunque altro tool grafico capace di salvare in ILBM.

Supponiamo di aver appena finito di scrivere un programmino, magari in Basic, e di volerlo dotare di un'icona un po' più fantasiosa di quella flow-chart-like che l'interprete crea automaticamente. Entriamo in Deluxe Paint, optiamo per il formato di schermo 640x200 con quattro colori per compatibilità con lo Workbench nel quale l'icona dovrà vivere e disegnamo ciò che vogliamo. Naturalmente non desideriamo che l'intera schermata diventi un'icona, ce ne basta una piccola parte; ritagliamo come brush (o pennello che dir si voglia) la porzione che ci interessa e salviamola attivando l'opzione Save nel menu Brush. Con Deluxe Paint abbiamo finito.

Lanciamo in esecuzione IFFConverter e attiviamo Load IFF brush; ci verrà chiesto il pathname completo del brush in questione, nella solita forma (ad esempio disco:file o disco:directory/file). Se il pathname è sbagliato l'AmigaBasic si ferma con una segnalazione di errore; bisogna allora dare un run per sbloccare la situazione. Se non ci sono errori il programma carica il brush in memoria e medita per un po' sulla sua struttura, poi ritorna alla schermata principale. Selezioniamo Save as icon; il programma richiede il pathname di un'icona preesistente, la cui immagine verrà sostituita con quella presa dal brush. "Ma non sarebbe più semplice creare un'icona da zero? A che scopo caricarne una preesistente?" si chiederà più d'un lettore. Il fatto è che un'icona è qualcosa di più di una semplice immagine. I file delle icone (quelli con il suffisso .info) contengono altre informazioni, come il tipo di oggetto da esse rappresentato (tool, project, drawer, garbage...), il tool di default (nel caso dei programmi Basic, il tool di default che viene automaticamente caricato in memoria clickando su di essi è l'interprete AmigaBasic) e svariate altre cosette. Sarebbe decisamente tedioso costringere l'utente a specificare ogni volta tutti questi parametri, quindi IFFConverter li estrae da un'icona preesistente.

Specifichiamo dunque il pathname del programma Basic di cui occorre cambiare l'icona; IFFConverter vi aggiunge automaticamente il suffisso .info se si accorge della sua assenza. Dopo il caricamento dell'icona e alcune ulteriori elaborazioni appare, per maggiore flessibilità, la richiesta del pathname con cui salvare la nuova icona, che nel nostro caso coincide con quello precedente, quindi ci limitiamo a battere l'ENTER e il gioco è fatto. In effetti ci sono dei casi in cui può essere utile che l'icona sorgente e quella destinazione non coincidano, per esempio se vogliamo dotare di icona un programma in Assembly o in C che non ne ha una di default come invece accade per quelli in Basic. In tal caso come icona sorgente andrà bene quella di un qualunque tool, ad esempio quella dell'interprete AmigaBasic, dal momento che le icone dei tool sono tutte più o meno equivalenti, al contrario di quelle dei project.

In determinate circostanze IFFConverter stampa messaggi di vario genere o richiede ulteriori informazioni; nel caso delle icone, ad esempio, se abbiamo fatto uso di una grafica con più di quattro colori il programma ci avverte che tutti i piani di bit dal terzo in poi saranno gettati via (attenzione che è inevitabile che l'immagine risultante possa a questo punto essere diversa dall'originale). La filosofia di base di IFFConverter consiste nel chiedere all'utente il minor numero possibile di informazioni, anche perché molte di esse sono di natura squisitamente tecnica e potrebbero generare richieste quasi incomprensibili, del tipo "quale valore devono assumere i flag di saveback, overlay e savebob?", per non parlare di problemi tecnici che richiederebbero strane manovre extra, del tipo "rientra in Deluxe Paint ed esegui un Remap per ottenere un colore di fondo nullo, poi riprova". In tutti questi casi IFFConverter cerca di utilizzare valori di default ragionevoli o di risolvere autonomamente i problemi tecnici. L'utente viene scomodato solo nel caso di una scelta critica, come ad esempio nel caso della distinzione tra VSprite e Bob (ma anche in questi casi battendo ENTER si accetta il valore di default).

Qualche nota sulle icone

FFConverter è stato progettato eclusi-

vamente per la creazione e alterazione di icone a immagine singola, cioè di quelle che non cambiano forma clickandoci sopra.

Se si desidera (com'è piuttosto verosimile che accada) realizzare un'icona a doppia immagine, la procedura è la seguente:
1) Creare le due immagini come brush separati, ma dotati delle stesse identiche dimensioni (conviene attivare l'opzione Coords dal menu Prefs mentre i brush vengono ritagliati).
2) Creare con IFFConverter due icone diverse con i brush così definiti.
3) Lanciare l'utility IconMerge presente negli Extras, il quale impone che le due icone abbiano le stesse dimensioni (ecco spiegato il punto 1). E' un peccato che IconMerge soffra di questa limitazione.

I programmatori in C o in Assembly desiderosi di iconizzare (piaciuto il neologismo?) i propri programmi facciano attenzione ad utilizzare uno header adatto all'attivazione da Workbench. Per l'Assembly, in particolare, sulle pagine di Amiga magazine è già apparso qualcosa del genere (IconHeader).

Un'ultima nota sulle icone: i normali colori dello Workbench sono, nell'ordine, blu, bianco, nero ed arancione, mentre quelli di default di Deluxe Paint (quando si usano solo quattro colori) sono nero, bianco, blu e rosso. Notate che blu e nero sono scambiati di posto. Onde evitare equivoci cromatici è opportuno scambiare tra loro questi colori nella paletta di Deluxe Paint prima di iniziare a disegnare icone.

Sprite per tutti i gusti

Abbiamo visto come cambiare un'icona; non stupitevi se, rientrando nello Workbench, non noterete alcun cambiamento, perché il sistema operativo non si è ancora accorto della modifica effettuata. Per costringerlo a prenderne atto è sufficiente chiudere e riaprire il drawer contenente l'icona alterata.

Passiamo adesso agli sprite, che possono essere di due tipi: VSprite (Virtual Sprite) e Bob (Blitter Object). La differenza è sostanziale, in quanto i primi sono veri e propri sprite, le cui immagini vengono visualizzate in sovrimpressione sullo schermo da otto sprite processor contenuti in Denise.

Ogni VSprite può avere tre colori più il trasparente, è sempre in bassa risoluzione ed è largo al massimo sedici pixel, mentre non esiste limite di altezza.

L'hardware di per sè sarebbe in grado di gestire un massimo di otto sprite, che devono avere a due a due gli stessi colori, ma queste limitazioni sono parzialmente aggirate via software con il ricorso agli sprite virtuali.

Le limitazioni effettive sono quindi le seguenti: non più di quattro sprite con colori diversi (oppure otto con gli stessi colori) sulla stessa riga orizzontale (ma sull'intero schermo possono esisterne dozzine). I VSprite utilizzano i colori da 16 a 31, di conseguenza sarebbe meglio usare al massimo 16 colori (da 0 a 15) per lo sfondo, onde evitare seccanti interferenze cromatiche. Il puntatore del mouse, ad esempio, è un VSprite.

I Bob sono fisicamente disegnati nella pagina grafica dal Blitter (e ciò richiede tempo), ma non hanno alcuna limitazione, eccetto quella (ovvia) di condividere la risoluzione e la paletta dei colori con lo sfondo sottostante (anche il numero massimo di colori deve coincidere), mentre i VSprite possono avere ognuno tre colori indipendenti scelti tra 4096.

Se ad esempio volete creare un Bob che dovrà essere usato in uno schermo 320x200 ad otto colori dovrete entrare in Deluxe Paint e selezionare tale modo grafico prima di iniziare a disegnare.

Come per le icone, una volta tracciato lo sprite bisogna ritagliarlo come brush e salvarlo (nota: se si desidera un VSprite occorre fare attenzione a non ritagliare una superficie larga più di sedici pixel. Aiutatevi con la lente d'ingrandimento). Lanciamo IFFConverter e selezioniamo prima Load IFF brush e poi Save as sprite; se il brush è piccolo e dotato di pochi colori il programma chiede se si desidera farne un VSprite, altrimenti non può che essere un Bob. Supponiamo di salvarlo con il nome di Sprite1; per riutilizzarlo in Basic, come illustrato nel manuale (pagina 8-92), dovremo utilizzare una serie di linee simili a queste:

```
OPEN"i",1,"df0:Sprite1"
OBJECT.SHAPE n,INPUT$(LOF(1),1)
CLOSE 1
```

dove n è il numero di identificazione dello sprite.

La paletta di colori

Quando Deluxe Paint salva un brush, include nel relativo file anche la paletta usata per disegnarlo. Se l'oggetto diventerà un VSprite si porterà dietro gli stessi tre colori che aveva al momento del salvataggio da Deluxe Paint e sarà poi visualizzato con quei colori indipendentemente dagli statement COLOR o PALETTE presenti nel programma che lo pilota; i Bob, al contrario, fanno propria la paletta dello schermo in cui vengono tracciati.

La conseguenza immediata di quanto detto è che se disegnamo una bellissima serie di Bob con una certa paletta di colori e poi li utilizziamo in un programma Basic che fa uso di una paletta diversa (come di regola accade, essendo i colori di default di Deluxe Paint e del Basic tutt'altro che coincidenti) l'aspetto dei Bob sarà del tutto diverso.

Onde risolvere questo fastidioso problema IFFConverter mette a nostra disposizione la funzione Save palette, che estrae la paletta dal brush appena caricato e la deposita in un file, il cui nome è scelto dall'utente, costituito da una serie di statement PALETTE in ASCII. Se volete utilizzare tale paletta nel vostro programma sarà sufficiente un MERGE con il suddetto file, ed in coda al listato appariranno gli statement PALETTE direttamente eseguibili che potrete spostare con un po' di Cut e Paste dove più vi aggrada (presumibilmente subito dopo le istruzioni SCREEN e WINDOW certamente presenti nel programma). Se per caso i brush sono stati creati con un tool grafico che NON include la paletta nel file (potrete accorgervi di ciò dal fatto che l'opzione Save palette non risulta abilitata) dovrete astenervi dal creare VSprite, e per quanto riguarda i Bob dovrete ricopiare la paletta manualmente. In realtà IFFConverter è capace di estrarre la paletta da una qualunque immagine IFF, anche da un'intera schermata (la codifica IFF dei brush e delle schermate è fondamentalmente identica), ma in tal caso andrebbe certamente in out of memory, a meno di non aumentare la RAM disponibile per il Basic con un poderoso CLEAR.

SCACCHI (e COMPUTER), SCIENZA O ARTE?

di Avelino De Sabbata

Si dice che gli scacchi siano il più nobile, nonchè uno fra i più antichi giochi con cui l'uomo si sia deliziato la mente e lo spirito. Introdotto in Francia all'epoca dei crociati, il gioco degli Scacchi venne praticato sopratutto dai persiani che lo avevano a loro volta importato dall'India nel VI secolo dopo Cristo durante il regno di Cosroe il Grande.

E' annosa la questione se sia preminente negli scacchi il carattere scientifico o quello artistico. Se si considera la capacità degli scacchi ad essere sottoposti ad una teorizzazione estremamente profonda, (come è di fatto dimostrato dal numero dei trattati teorici a questo gioco dedicati, di gran lunga superiore al numero dei testi scritti su tutti gli altri giochi messi assieme), si ha la netta sensazione che negli scacchi predomini nettamente il carattere scientifico. Se invece ci si avvicina agli scacchi come ad un campo di conquista che permette al pensiero di creare opere di particolare valore estetico, allora si ha la sensazione che negli scacchi predomini il carattere artistico.
Queste diverse valutazioni, sono infatti molto soggettive, e dipendono totalmente, come si è visto, dal modo in cui ognuno si accosta al gioco, scienza o arte che sia.

Una macchina pensante

Fin dal Medio Evo il sogno di creare una macchina pensante a propria immagine e somiglianza, ha assillato l'uomo. E' emblematico il fatto da tutti conosciuto dell'immenso imbroglio messo in opera nel 1769 da quel gran millantatore che fu Wolfang von Kempelen, creatore del famoso ed allora strabiliante "Giocatore di scacchi turco", una macchina costituita da un tavolo ed un uomo meccanico, la quale ben celato tra i congegni palesemente in mostra, nascondeva al proprio interno un nano, bravissimo giocatore a scacchi!

Col passare dei secoli e con l'accumularsi della conoscenza, l'alchimista di allora, si è trasformato nel matematico e nel fisico del nostro secolo, e pur messa da parte la pretesa di dar vita ad una creatura a propria immagine, non ha abbandonato la speranza di costruire macchine che avessero almeno la capacità di pensare come lui, e mai come ora, con la comparsa dei computer, questo obiettivo gli è sembrato così vicino.

Le caratteristiche intrinseche del gioco degli scacchi, (prevedibilità di tutte le mosse, conoscenza completa, da parte dei contendenti, della situazione sul campo, risultato finito di ogni partita), hanno fatto si che questo si sia rivelato, fin della nascita dei primi computer, uno stimolante soggetto di studio e ricerca dei più impegnati nella programmazione. Certamente esistono numerosi altri giochi che possiedono queste caratteristiche, ma gli scacchi sono unici nel presentare una estrema complessità nelle combinazioni tra posizione sulla scacchiera e mosse possibili, in modo da rendere pressochè impossibile, nella pratica, il fatto che si possa giocare due partite uguali. A questo riguardo, è stato calcolato che partendo dalla posizione iniziale dei pezzi sulla scacchiera, effettuando sempre mosse valide, è possibile raggiungere un numero di posizioni diverse che assomma alla strabiliante cifra di circa 10 alla 46a.

Arrivano i primi programmi

I primi programmi che "sapevano" giocare a scacchi, ancora grezzi e con un'euristica non molto avanzata, erano chiaramente appannaggio di quei pochissimi Santoni che potevano avere accesso ai mastodontici calcolatori, relegati nelle asettiche sale di università e grossi centri di calcolo.

Si dovrà aspettare fino al 1976 perché una piccola azienda che operava nel cam-

po degli elaboratori, la Fidelity Electronics LTD. di Miami in Florida, grazie al fiuto di un proprio dipendente, riesca a proporre al mercato un piccolo elaboratore il cui contenitore non era altro che una scacchiera adatta all'interfacciamento col computer in essa contenuto. Fu un immediato successo, nonostante i macroscopici difetti di quella prima scacchiera elettronica: scarsa abilità nell'analisi di gioco, impossibilità di eseguire l'arrocco, mancata promozione dei pedoni in ottava casella ed altro ancora.

Con il passare del tempo, nei pochi anni che ci separano da quella data, grandi progressi sono stati ottenuti sia dalla Fidelity che da altre ditte nate sulla scia di questa, e la possibilità di produrre micro ed home computer con una ragguardevole memoria interna ed a prezzi di grande mercato, hanno dato la possibilità alle Software House di offrire un grande numero di programmi per il gioco degli scacchi, per ogni tipo di computer.

Ovviamente, è chiaro che per quanti

progressi potranno essere fatti sia dalla tecnologia costruttiva dei computer, sia dalle tecniche euristiche impiegate negli algoritmi di calcolo dei programmi, non sarà mai possibile creare una macchina che possa offrire le stesse emozioni che un giocatore prova affrontando un avversario in carne ed ossa. Questi tipi di programmi, hanno però dalla loro il fatto di essere degli ottimi insegnanti per chi è alle prime armi, oppure degli instancabili allenatori e flessibili strumenti per lo studio e l'analisi, offrendo alcune possibilità che un avversario umano ed una scacchiera normale non potrebbe mai offrire.

Gli scacchi ed il computer nelle nostre case

C'è da sapere che io sono da sempre stato affascinato dal gioco degli scacchi, pur essendo un giocatore poco più che mediocre. Alcuni anni fa mi comperai una scacchiera della Fidelity, un mostro con il quale già ai primi livelli c'è da sudare le classiche sette camice per riuscire a mantenere una difesa accettabile. Un'altra passione oltre agli scacchi, (più recente questa però), è il computer. E' così che dopo aver pasticciato su di un 64 prima, ed un 128 poi, ho affannosamente raggiunto uno smagliante traguardo: l'AMIGA!

Ebbene, ho subito voluto provare i programmi di scacchi esistenti per questo computer, ma ne sono rimasto relativamente deluso, più che altro per la lentezza di gioco che questi offrono. Come di certo saprete è possibile attualmente reperire per Amiga il famoso Chess Master 2000, a cui si è aggiunto da qualche tempo il programma The ART of CHESS, di cui è stata fatta una recensione sul primo numero di Amiga Magazine. E' inoltre di re-

centissima apparizione sul mercato "parallelo" un'ulteriore software: Battle Chess, ed è su questa opera che mi voglio soffermare, e più oltre capirete il perché.

Alla scoperta di Battle Chess

Qualche giorno fa un amico con cui parlavo, e che conosce le mie "tendenze", mi offrì di provare questo nuovo programma per giocare a scacchi, senza volermi anticipare niente: tieni disse, e vedi un po' cosa te ne pare. Vabbè, penso, alle solite lo provo e lo metto nel cassetto, io alla mia scacchierina non ci rinuncio. Giunto a casa, infilo il dischetto nel drive e mi dispongo in attesa, (a proposito che lagna questi drive!). Una prima gradevole sorpresa mi giunge con la splendida schermata di presentazione, eh già, penso, ho proprio una bella macchina!...

Dopo ancora qualche ronzio del drive, finalmente questo tace, e sullo schermo mi viene presentato il campo da gioco: una scacchiera dal pavimento marmoreo, vista in prospettiva. I pezzi, dalla foggia singolare, ma disegnati mirabilmente, già disposti sulle case di partenza. Bello, molto ben disegnato, peccato che per giocare seriamente una scacchiera così crei un po' di confusione, anche perché il cavallo diviene distinguibile dal pedone e dall'alfiere solo con un poca di pratica, essendo tutti e tre i pezzi rappresentati da fi-

gure umane(?), soltanto vestite in modo differente. Bene dico, andiamo avanti, vediamo cosa ci offrono i menù. Impugno il mouse e premo il pulsante destro e, altra sorpresa, appare la barra dei menù a foggia di lunga fiamma, (avete presente quelle bandiere strette e lunghe?), mentre ogni menù viene srotolato da due graziosi put-

ti che lo sorreggono, sbattendo con energia le loro piccole ali. Notevole l'effetto sonoro di tutto quello... starnazzare!

A questo punto, prendo mentalmente nota di quello che offrono i quattro menù: (Disk, Move, Settings e Level), ed inizio a giocare, realmente divertito ed incuriosito al tempo stesso. Ho infatti già la sensazione di avere a che fare con qualche cosa che non rientra nei canoni stabiliti dai consueti programmi di scacchi.

Posso osservare che è molto facile, anche per chi non conoscesse le regole degli scacchi, trovare le mosse possibili. Ogni casa su cui è disposto un pezzo che è possibile muovere, viene evidenziata quando il puntatore del mouse viene a trovarsi in corrispondenza di essa. Non rimane altro da fare che selezionare la casella contenente il pezzo che vogliamo muovere.

Da questo momento in poi, verranno evidenziate, (durante gli spostamenti del mouse), solo quelle case in cui è possibile portare il pezzo selezionato. Un ottimo approccio per chi si avvicina per la prima volta agli scacchi: imparare facendo, è una delle migliori regole che si possano applicare a qualsiasi branca della didattica.

Dopo aver selezionato la mossa, (avanzo il pedone di re di due case), mi aspetta un'ulteriore sorpresa. Il movimento che ho deciso da far fare al mio pedone, viene eseguito per mezzo di una efficace animazione, accompagnata con tanto di effetti sonori: scalpiccio di passi e clangore di armature. Semplicemente spettacolare!

Continuando nel gioco, posso osservare che ogni pezzo ha un suo particolare modo di incedere, ed ognuno con un accompagnamento di rumori diversi. Il pedone avanza con atteggiamento marziale ed un gran sbattere di tacchi, il cavallo si fa strada tra gli altri pezzi con rumorosa e maleducata irruenza, la regina si muove ancheggiando con silenziosa ed altera maestà regale, il re per contro, trascina stancamente le sue vecchie membra con uno sforzo che sembrerebbe ogni volta essergli fatale.

Ad ogni movimento delle torri, queste invece si trasformano in possenti titani di pietra che avanzano lenti e pesanti, facendo rintronare ad ogni passo il campo da gioco di un assordante fragore.

La partita (bene o male) prosegue, e giunge l'ora di effettuare qualche cambio, tanto per alleggerire la scacchiera. Con un mio pedone, prendo un pedone avversario, e qui, (non dite che sono monotono), si rinnova e si moltiplica la sorpresa. Ma certo, dovevo immaginarlo dal nome del programma! Ecco che i due pedoni contrapposti si predispongono al combattimento, e dopo alcuni colpi, (con un impressionante contorno di lamenti ed imprecazioni), il pedone preso si accascia al suolo sconfitto, ed i suoi resti si dissolvono nel nulla!

Da questo momento, il gioco perde anche quella poca seriosità che gli avevo imposto: per me non è più una partita a scacchi, in quanto le mosse che faccio hanno l'unico scopo di contrapporre i pezzi nemici per scoprire come essi combattono, colpiscono, offendono, in un'orgia di esilarante violenza simulata. Tra le mazzate, colpi bassi, geniali trovate ed efferatezze di ogni genere, non riesco a trattenere qualche risata di gusto, al che incuriosite, mia moglie e mia figlia si aggregano a godere del sano spettacolo offerto da questa accozzaglia di BIT, palpitanti nella facinorosa vita che anima i personaggi di Battle Chess.

L'esplorazione ci svela che ad ogni coppia di avversari diversi, corrisponde una determinata sequenza di animazione, ogni pezzo cioè si comporta in una data maniera in relazione al pezzo avversario che si trova di fronte.

Ma non è tutto, l'animazione cambia anche in relazione al pezzo che risulta perdente (o vincente!). Così un pedone, ad esempio, si comporterà in un determinato modo per prendere un alfiere, e nella sequenza contraria, quando cioè l'alfiere prenderà un pedone, assisteremo ad una scena del tutto diversa.

Qualsiasi descrizione non renderà merito della puntigliosa cura con cui sono state create le animazioni ed i suoni di questo gioco, (basti pensare che gli effetti sonori campionati, occupano la bellezza di 400 kbyte sul dischetto!).

Bè si chiederà qualcuno, e che ne è rimasto di tutta la filosofia, scienza ed arte racchiuse nell'immortale gioco degli scacchi? Come si fa a giocare a scacchi con un dissacrante aggeggio del genere? Questa è infatti la critica maggiore che si può muovere alla Interplay, anche se scorrendo i menù notiamo la possibilità di selezionare una scacchiera bidimensionale come campo da gioco, che si rivela nettamente più professionale, chiara e veloce di quella a tre dimensioni. Con il campo da gioco in 2d infatti i pezzi sono disegnati nel modo classico, e vengono eliminate le animazioni, principale causa del notevole rallentamento del gioco in 3d, dovendo essere di volta in volta caricate da disco.

Osservando la questione da un'ottica diversa potremmo invece dire che non manca un certo approccio in qualche modo genialoide, nel presentare un programma per il gioco degli scacchi 'confezionato' in questa maniera. E mi riferisco alla prerogativa che Battle Chess possiede di certo rispetto agli altri programmi suoi simili, che è quella di attirare verso questa meravigliosa scienza (o arte?), anche chi, o per un motivo o per l'altro, non si era mai avvicinato.

Questo articolo, che inizialmente voleva pretestuosamente riferirsi a Battle Chess solo come occasione per parlare di scacchi, è di fatto diventato una recensione di questo programma. Vediamo quindi di portarla a termine, prendendo in esame anche le altre caratteristiche del software in questione, anche se a mio parere, lo svolgimento dei combattimenti sulla scacchiera tridimensionale, è l'aspetto più apprezzabile di tutto il programma.

Menù di Battle Chess

Dal menù DISK, possiamo caricare (LOAD GAME), una partita già giocata, oppure salvare (SAVE GAME), la partita in corso.

L'opzione NEW GAME ci permette in ogni momento di iniziare una nuova partita, naturalmente perdendo la situazione attuale. Molto utile l'opzione successiva (SETUP BOARD), tramite la quale avremo la possibilità di definire una certa scoperta, naturalmente con il software originale, appena questo si renderà disponibile. Battle Chess è prodotto dalla Interplay/Electronic Arts.

P.S. Al momento della stesura dell'articolo, la situazione si rivelava come riportato dall'articolista. Adesso, al momento d'andare in stampa, BattleChess viene distribuito sia dalla LAGO che dalla CTO.

l'AMIGA é anche fotogenico ?

l'AMIGA é anche fotogenico ?

di Giorgio Dose

Una buon lavoro grafico merita senz'altro una buona fotografia. E' indubbio inoltre che Amiga con i suoi 4096 colori e la sua superba risoluzione rappresenta un eccellente soggetto. Molte persone interessate alla grafica ed all'Amiga non pensano a quante bellissime e importanti fotografie dello schermo potrebbero fare. Dopo tutto esse ormai appaiono ovunque sui giornali, pubblicità, riviste e non esiste prodotto software che non si avvalga di qualche riproduzione delle sue schermate per farsi un po' di pubblicità.

Cosa serve per cominciare

L'articolo non è un trattato di fotografia e si rivolge a coloro che hanno una certa familiarità con la macchina fotografica oltre a conoscere la funzione dei dispositivi più comuni come il diaframma, il tempo di esposizione, ecc. (in caso contrario si consiglia di portare l'Amiga dal fotografo di fiducia ... hi! hi!).

La macchina fotografica deve essere una reflex da 35 mm; può essere manuale o automatica ma è preferibile un'automatica che consenta anche l'esposizione manuale con tempi di almeno due secondi. Il cavalletto è d'obbligo naturalmente e la pellicola più adatta è quella che si usa comunemente con una sensibilità di 100 ASA.

Usando l'obiettivo standard da 50 mm le immagini risulteranno inclinate o convesse molto di più di come appaiono sul

Foto 1

Foto 2

Foto 3

monitor. Per una visione più professionale (ma anche più costosa) e per avere un'immagine piatta bisogna servirsi di un teleobiettivo medio o, ancor meglio, di uno zoom.

La distanza ottimale di posizionamento della macchina è circa sui tre metri dal monitor ma può essere aumentata in funzione della focale dell'obiettivo; lo zoom deve venir regolato, per ottenere un'immagine più piatta possibile, alla massima focale e in modo che lo schermo occupi tutto il fotogramma. Una cosa molto importante è che l'asse dell'obiettivo sia esattamente sullo stesso piano del monitor onde evitare la deformazione dell'immagine.

Foto 4
Esposizione normale

Accorgimenti e regolazioni

Le condizioni in cui prendere la foto sono senz'altro il problema cruciale per ottenere dei buoni risultati. Avete mai visto delle foto perfettamente a fuoco e con dei colori smaglianti macchiate da bagliori o strani riflessii di luce? In questo caso probabilmente il fotografo ha commesso l'errore di prendere le sue foto con qualche porta o finestra aperta o con il monitor vicino a qualche fonte di luce.

Quando si fotografono delle parti scure con delle superfici riflettenti come il nostro monitor è molto importante evitare che l'immagine sia colpita da luce diretta. La stanza deve essere oscurata ma non completamente buia, bensì illuminata da una luce fioca che proviene sempre da dietro al monitor.

Il cavaletto va regolato in modo che il centro del monitor e l'asse della macchina fotografica si trovino sullo stesso piano orizzontale. Onde evitare deformazioni la superfice del monitor deve essere perfettamente verticale e si fa notare che normalmente così non è; se il monitor è sul suo normale piano d'appoggio, la superfice dello schermo risulta leggermente inclinata all'indietro. E' quindi bene, volendo ottenere dei risultati perfetti, provvedere a sollevare il monitor sul retro.

A questo punto si regola lo zoom per la massima focale, si mette perfettamente a fuoco, cioè fino a vedere chiaramente le linee ed i pixel dello schermo, quindi si zumma un po' indietro per ottenere nel mirino l'immagine che si desidera fotografare, e che può anche non essere l'intero schermo.

Anche se possedete una macchina automatica non lasciatevi prendere dalla frenesia di scattare. Per ottenere una buona foto sarà prima indispensabile effettuare una regolazione che potrebbe apparire un po' strana e che consiste nell'abbassare la luminosità ed il contrasto dello schermo. Scattando con uno schermo regolato normalmente le zone dell'immagine a luce più intensa risultano diffuse, con i contorni poco definiti, come circondate da un alone luminoso.

Vi chiederete di quanto vanno abbassati il contrasto e la luminosità. Non si pos-

Foto 5
Esposizione + 1 diaframma

Foto 6
Esposizione - 1 diaframma

sono dare dei valori precisi perché essi dipendono dalla luminosità media dell'immagine che si desidera riprendere; si possono solo fornire delle indicazioni di massima ma vedrete che con un minimo di esperienza, dopo aver esaminato i primi scatti, sarete in grado di valutare in modo ottimale la giusta luminosità e contrasto. Comunque tenete presente che essi vanno ridotti fino a ridurre lo sfarfallio delle zone più luminose ma non tanto da alterare i colori o perdere qualche dettaglio dell'immagine.

Finalmente si scatta

Non resta a questo punto che regolare il valore del diaframma e il tempo di esposizione. Il valore del diaframma non è molto importante perché essendo lo schermo una superfice piatta non esiste il problema della profondità di campo e possono andar bene anche le massime aperture.

Volendo essere pignoli, però, è bene usare i valori intermedi in quanto con essi l'obiettivo ha sempre una resa migliore in termini di definizione.

Quello che invece risulta importante è il tempo di esposizione che deve essere sempre uguale o più lungo di 1/25 di secondo.

Questo perché lo schermo di Amiga viene tracciato, nel caso dell'alta risoluzione interlacciata, esattamente 25 volte al secondo.

Se il tempo di scatto è più rapido l'immagine risulterà non uniforme o incompleta. I migliori risultati si ottengono con tempi che scendono sotto l'ottavo di secondo e che arrivano fino a uno o due secondi.

Rispettando le indicazioni di cui sopra ed osservando l'esposimetro interno alla macchina regolate, a questo punto, il diaframma ed il tempo per una corretta esposizione.

La maggior comodità d'uso si ha con una macchina dotata di esposimetro a led che, nelle condizioni di scarsa luminosità in cui siamo, risulta essere ben visibile e consente una facile regolazione.

E' possibile naturalmente lavorare anche in automatismo ma, in questo caso, controllare che il tempo di esposizione rientri nei valori indicati.

Ricercando la perfezione

Operando come fin qui illustrato si otterranno senz'altro delle buone foto. Se però vogliamo superare noi stessi e raggiungere dei risultati perfetti dovremo ricorrere alla seguente tecnica.

Non sempre con un'esposizione normale e secondo quanto indicato dall'esposimetro si ottengono le immagini migliori. A volte una leggera sovraesposizione o sottoesposizione fornisce risultati insperati.

Per ottenere una foto sicuramente eccezionale sarà quindi necessario riprendere la stessa immagine più volte e scegliere poi quella migliore. Il numero dipende dall'esperienza che col tempo vi sarete fatti e dalla quantità di pellicola che siete disposti a sacrificare. Se non vi sviluppate da soli le vostre foto vi consigliamo di cominciare usando pellicola per diapositive che essendo trattata in modo standard consente una migliore valutazione delle differenze tra le varie esposizioni.

Una sequenza di cinque esposizioni è normalmente sufficiente. Esse dovranno essere riprese nelle seguenti condizioni:

1a: diaframma più aperto di due valori
2a: diaframma più aperto di un valore
3a: esposizione normale
4a: diaframma più chiuso di un valore
5a: diaframma più chiuso di due valori

Vediamo con un esempio cosa ciò significa. Se, per una determinata immagine, l'esposimetro fornisce, per un'esposizione normale, un valore di diaframma 5.6 ed un tempo di esposizione di 1/2 sec, tenendo fermo il tempo a 1/2 secondo, scatteremo cinque foto rispettivamente con diaframma impostato ai valori 2.8, 4, 5.6, 8 e 11.

Volendo, invece di variare il diaframma, si può tenere quest'ultimo costante e variare il tempo di esposizione. Nel caso dell'esempio appena descritto si effettueranno cinque esposizioni con diaframma 5.6 e tempi di 2 sec, 1 sec, 1/2 sec, 1/4 sec, 1/8 sec.

Nel caso si operi con una macchina automatica, questa di solito è predisposta per effettuare delle variazioni all'esposizione. Generalmente si lascia il diaframma costante e si agisce sulla regolazione della sensibilità della pellicola su dei valori appositamente contrassegnati con +2, +1, 0, -1,-2. Logicamente questo porterà a una variazione del tempo automatico di esposizione. Assicurarsi solamente che esso non sia più breve di 1/25 di secondo.

Sviluppo ed ingrandimento

Quando portiamo le nostre foto a sviluppare esse vengono trattate in modo automatizzato e non è possibile chiedere dei trattamenti personalizzati a meno di non voler affrontare notevoli spese.

Per coloro che amano sviluppare da soli i propri film, si consiglia, volendo fare degli ingrandimenti, di sviluppare il film per un'alta densità.

L'incremento di densità rende le stampe più contrastate e con i colori più vivi e decisi. La densità normale tende invece a produrre delle foto piuttosto chiare con dei colori meno intensi.

Alcuni esempi

Parlando di fotografie molto spesso una sola immagine vale più di molte parole. Ecco perché abbiamo voluto illustrare quanto fin qui descritto.

La prima foto, rappresenta una scacchiera ed è senz'altro un esempio da non imitare; in essa sono condensati tutti gli errori più comuni che si possono fare. La foto è stata scattata a mano libera con un obiettivo da 50 mm. e infatti si nota la convessità dello schermo e la centratura non perfetta. Poi, come se non bastasse, sull'immagine sono presenti dei riflessi dovuti ad una finestra non oscurata.

Seguono alcune foto scattate con un'esposizione normale e che risultano comunque molto buone.

Per ultimo abbiamo presentato tre immagini del famoso babbuino, riprese con la tecnica di variazione dell'esposizione appena descritta.

E' evidente che disponendo di più foto riprese con diverse esposizioni è molto più facile ottenere una foto perfetta anche se ciò comporta un maggior dispendio di pellicola.

113 P

SCOPRILO FINO ALL'ULTIMO BIT

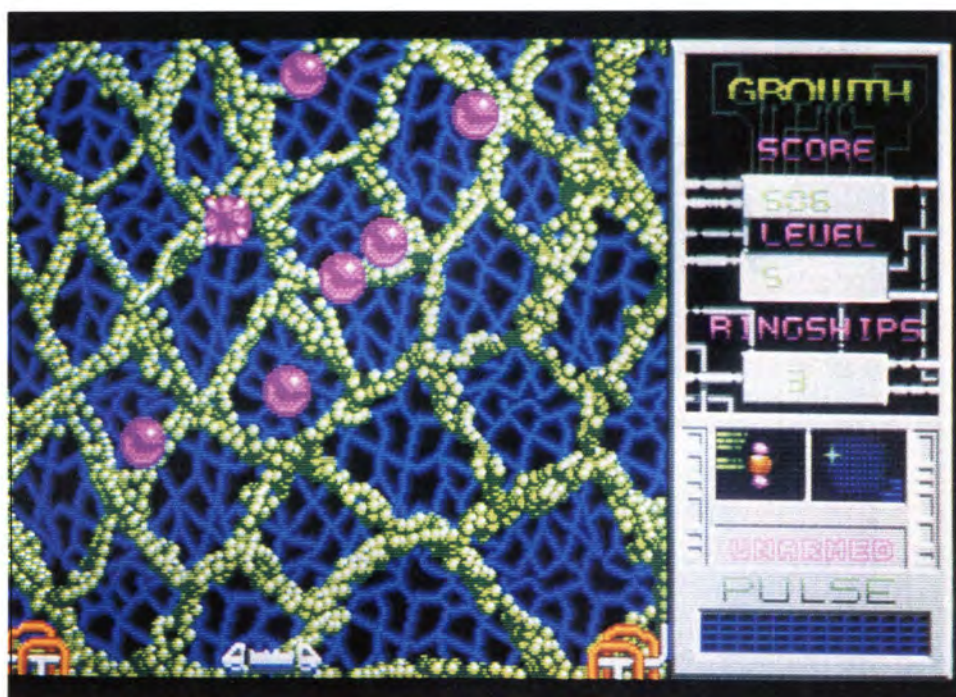
Scopri tutto quello che può darti il tuo computer AMIGA.
Ogni mese, dal GRUPPO EDITORIALE JACKSON, tre riviste che ti danno di più: AMIGA TRANSACTOR per i programmatori più smaliziati, AMIGA MAGAZINE per chi vuole conoscere il proprio computer sempre di più e AMIGA MAGAZINE GAMES per i giocatori più accaniti.
AMIGA TRANSACTOR, AMIGA MAGAZINE e AMIGA MAGAZINE GAMES: un panorama completo ed esauriente dell'universo di AMIGA, tre riviste per usare il tuo computer fino all'ultimo bit.

AREA CONSUMER

Scegli il meglio: scegli Jackson

TOP GAME

GROWTH

Salvatevi dall' attacco DEGLI ALIENI

Schermata di GROWTH

di Alessandro Prandi

Dopo essere penetrati nel centro vitale del maligno Dreyficus avete la possibilità di bloccare la crescita aliena che sta annichilendo il mondo. Dovete sopravvivere ai 25 livelli del nefando tumore per confrontarvi nella battaglia finale contro l'odioso Dreyficus.

Circoscrivendo l'escrescenza e distruggendo le cellule in riproduzione potrete attaccare e distruggere il nucleo del tumore. Il gioco si svolge attraverso 25 livelli di difficoltà che in un entusiasmante crescendo vi prepareranno alla lotta finale. Per spostatare e controllare il vostro mezzo d'assalto usate il mouse, ricordate solo

Mouse a sinistra	**- muove in senso antiorario**
Mouse a destra	**- muove in senso orario**
Tasto sinistro	**- fuoco**

La distruzione dell'escresenza deve avvenire prima che questa intacchi le pareti dello schermo, altrimenti perderete una vita delle tre a vostra disposizione.

Per evitare la riproduzione delle cellule infette dovete sparare finchè non riuscite a crearvi un varco che permetta ai vostri proiettili di giungere al cuore centrale. Mentre cercate di annientare queste forze malvage fate attenzione a dei corpi di colore blu, questi vi permetteranno, una volta colpiti, di distruggere aree più vaste.

Man mano che proseguite attraverso i vari livelli, la vostra missione diventa sempre più difficoltosa e il contatto con le cellule aliene diviene sempre più probabile, ad un certo punto la destrezza diventa un'arma indispensabile.

Ogniqualvolta distruggete successivamente 2 livelli del tumore vi troverete a confronto con una diversa configurazione dell'escrescenza.

Nella prima configurazione dovete distruggere completamente le cellule e il nucleo centrale, ma allo stesso tempo è indispensabile mantenere in gioco una capsula bloccandola con il vostro aereo, in modo che non possa uscire dallo schermo.

Nella seconda configurazione dovete distruggere le capsule del tumore prima che raggiungano la parte inferiore dello schermo.

Se riuscirete a completare successivamente tutto ciò, arriverete allo scontro finale con il tremendo Dreyficus.

Schermata di GROWTH

VETRINA

Dall'Oriente all'Europa

Ofra Haza, la voce che viene dall'Oriente, è approdata in Europa con le sue due ultime incisioni che sono le prime che hanno varcato le soglie del suo Paese d'origine, Israele, nel quale da tempo gode di una notevole celebrità. Una veloce presentazione: Ofra Haza ha trent'anni, vive a Tel Aviv dove ha già pubblicato 18 album, la famiglia di origine è yemenita, una bellezza tipicamente orientale ed una voce appassionata e sensuale.

Gli ultimi suoi due album (Yemenite Songs e Shaday) sono stati creati appositamente per entrare in un mercato più vasto quale quello internazionale. Questa scelta commerciale ha inevitabilmente inciso sulla qualità e sulle modalità musicali e vocali delle canzoni presentate pur senza annullarne completamente il fascino. Dei due album il primo (reperibile solo nei negozi d'importazione) è quello che meno ha risentito di queste mediazioni tra le due culture (Orientale ed Occidentale) ed infatti contiene anche alcuni frammenti dei poemi di Rabbi Shalom Shabazi (leggendario poeta yemenita del Seicento) trasportati in musica.

In questo disco la musica mantiene ancora tutto il suo fascino mediorientale con l'uso dei ritmi tipici e di strumenti originali, con solo una sfumatura di arrangiamenti più moderni. La voce emana tutto il suo calore e la sua magicità grazie anche all'uso delle lingue originali (l'ebraico, l'arabo e l'aramaico). Il suo secondo album (in vendita anche in Italia) risente di più della contaminazione della disco dance e della musica pop, la voce viene inevitabilmente alterata dalla pronuncia di parole inglesi.

Nonostante tutto, le manipolazioni su questo album non fanno perdere completamente il fascino originale di questa nuova stella della musica internazionale anche perché Ofra nei mesi scorsi, malgrado le pressioni dei discografici, si è rifiutata di presentare al suo pubblico un album da lei ritenuto troppo commerciale e, dopo averlo buttato senza troppi ripensamenti, si è messa al lavoro per incidere i brani di Shaday. Comunque, tralasciando ogni possibile critica sulla correttezza o meno di europeizzare alcune melodie proprie del suo Paese, Ofra Haza rimane un personaggio tanto più meraviglioso quanto più si avvicina alla sua forma musicale originale.

Un pizzico d'Irlanda

Gli U2 raggiungono il vertice della loro avventura nel mondo del rock con l'album del 1987 "The Joshua Tree" che segna l'inizio di un successo che varcherà i confini inglesi ed europei. Il rock viene vissuto dai quattro ragazzi di Dublino che poi formeranno gli U2 come la via per fuggire da una Irlanda troppo bigotta e tradizionalista. La loro avventura attraverso la musica rock risale all'inizio degli anni Ottanta ed oggi gli U2 sono, per alcuni, la più grande rock'n'roll band contemporanea e ritengono il loro recente album "Rattle And Hum" la celebrazione più evidente di tale

primato.

Questo nuovo capitolo della discografia degli U2 è una mescolanza di brani inediti e di brani vecchi registrati in parte in studio ed in parte dal vivo e che in alcuni momenti pare voglia simboleggiare la consacrazione di un mito. Tra i brani vecchi risultano alcune delle icone del sacro rock'n'roll quali "Helter Skelter" dei Beatles, la parte chitarristica di Jimi Hendrix nel brano "The Star Spangled Banner" e "All Along The Watchtower" di Bob Dylan. L'uscita di questo album con la congiunzione degli stati generali del rock passati riproposti sia sul disco che nella loro ultima tournee americana, il volere essere trasgressivi ad ogni costo e le crisi religiose rendono ancor più unici gli U2.

Questo album è comunque da considerarsi un album di transizione dove il gruppo tenta di collegarsi alla grande classicità rock dei Dylan e dei bluesmen staccandosi dal folklore celtico da cui provengono e liberandosi dai retaggi punk che hanno costituito per vari album il continuum con la scena dalla quale sono emersi per poi ripartire verso nuove esperienze. In definitiva gli U2 non possono essere considerati avanguardia allo stesso livello dei Doors o di Hendrix o di Lennon; ciò nonostante essi possono essere considerati credibili in quanto sintetizzano atteggiamenti ed istanze che la comunità rock si porta dietro come eredità e il loro fascino, e anche quello di questo loro ultimo album, sta anche nel non voler abbandonare i messaggi di un passato ancora vivo.

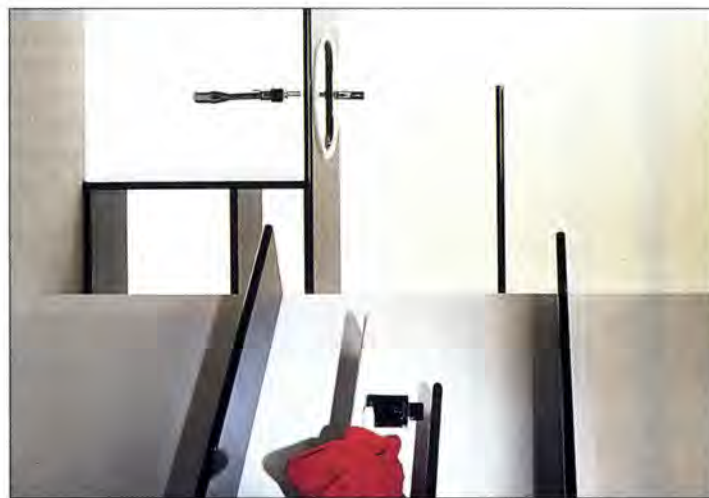
Una storia di amore e di morte

Un affare di donne (Une affaire de femmes, 1988), di Claude Chabrol, con Isabelle Huppert, Francois Cluez, Nils Tavernier, Marie Trintignant.

In questo film il regista Chabrol propone alcuni dei suoi temi prediletti e cioè la provincia francese con la sua ipocrisia e la sua amoralità associata alla triste situazione della Francia nel periodo dell'occupazione tedesca. La trama del film è tratta da un fatto realmente accaduto in Francia nel '43: il 31 luglio di quell'anno fu eseguita la sentenza di morte che un Tribunale Speciale aveva emesso contro Marie Giraud condannata per aver procurato aborti, uno dei quali aveva causato la morte della paziente. Una sentenza così pesante trova una sua giustificazione se rapportata al periodo in cui fu emessa: nel 1943 nella Francia di Vichy era forte l'esigenza di condanne esemplari per dimostrare la saldezza dei valori morali e di amor patrio del Paese e, l'inflessibilità della punizione per coloro che in qualche modo cercassero di minare questa sicurezza interna.

In una tale situazione non era difficile vedere in una procuratrice di aborti una sovvertitrice dei valori morali, un pericolo per l'istituzione familiare e, conseguentemente, una nemica della patria. Per tutta questa mescolanza di fattori, e non per la sua colpa reale, Marie Giraud fu ghigliottinata, fu comunque una delle ultime donne francesi a subire una tale sorte. Nel film Isabelle Huppert è Marie, la quale con grande bravura presenta questo personaggio dalla connotazione fortemente negativa in una donna comune, in una madre dolce ed affettuosa, determinatamente decisa a sopravvivere, istintivamente amorale, con la passione per il canto ed, a causa degli eventi, avida e faccendiera.

La vita di Marie è difficile: il marito in guerra, due figli da mantenere e la povertà del periodo bellico da combattere. In questa atmosfera la donna si trova ad aiutare un'amica ad abortire; a questa prima pratica di aborti ne seguono altri e la cosa, casualmente, diventa un'attività redditizia che la porta ad arricchirsi velocemente e, neppure la morte di una paziente sembra scalfire questo nuovo stato di benessere.

Il ritorno del marito, malato, depresso e moralmente vinto, non è gradito a Marie che lo disprezza.

Successivamente diventa l'amante di un collaborazionista dei tedeschi ed inizia a prendere lezioni di canto. Il marito diventa sempre di più un peso e la donna è disposta a pagare una cameriera affinchè ne diventi l'amante pur di sfuggire alle sue attenzioni. Questa situazione diventa insostenibile per tutti e due finchè il marito, esasperato, la denuncia anonimamente per le sue pratiche abortive e per la morte, durante una di queste, di una cliente. Da questa denuncia ne derivano il processo e la conseguente condanna a morte, alla quale Marie reagisce violentemente tanto le appare assurda ed ingiusta.

Nel complesso l'ambientazione storica è molto curata e d'effetto, anche se il film appare in alcune sequenze piatto ed in altre, specialmente nella seconda parte, piuttosto frettoloso.

Medium density

New density è il materiale di cui sono composte le nuove mensole della ditta Nicom. Sono disponibili nei colori bianco e nero in modo da permettere un facile accostamento di colori. Queste mensole di misura diversa e di solida fattura sono resistenti all'umidità e si prestano ad un uso molteplice: da ripiano per incasso del lavabo a portafaretti o altro. Per quanto riguarda i supporti, questi possono essere a vista o a scomparsa ma sempre di facile montaggio. Questo accessorio per la casa è prodotto dalla Nicom di San Paolo d'Argon (Bergamo), tel. 035/958224.

La radio dei lupi di mare

Questa radio a batteria dalle ridotte dimensioni, non supera infatti i 15 centimetri di diametro è resistente all'acqua. Per facilitarne il montaggio si sono previsti tre diversi attacchi e per accenderla dovrete premere il grosso bottone centrale. Il costo di tale apparecchio è di L.70000 e viene distribuito da Outlook Zelco di Presezzo (Bergamo, tel. 035/611562.

Un altro essere vivente , non sai che gioia mi fa incontrarti...
EVVIVA !!
PLUM!
PAF
Eh ?
?!
Accidenti la sventola !! Ma che ho fatto di male ?
Che tipo...
Ho capito , ci vuole tatto...ehm...
SNAP!

Qualche ora dopo...
...e cosi io bla, bla, bla...
i miei colleghi mi cercheranno ma il mio veicolo...
...e stato divorato da un "coso" verde...
...non ti dico il terrore...roba da infarto...ehi, che bel posticino qui !

Facciamo un altro tentativo
e' tardi Mike...
LO SO !
almeno non moriro di fame stasera
wow!
uh? che cos e?
Un disegno ?!
CRIC CRIC
...farebbe la gioia di uno storico
Oh ! Ho capito ,un Graffito storico che riporta gli avvenimenti piu importanti del villaggio e della tribu...
...cosa che io non sono !

...e poi devo
solo aspettare
che vengano
a prendermi !

uhm...chissa' com'e'
quel brodino...
quasi quasi...

Uh? Per me?
Ehi grazie,
come hai fatto
a capirlo
figliolo?

Che brutto colore ha,
chissà il sapore...
GLOB!

CONTINUA

ABBONAMENTO JACKSON = SERVIZIO COMPLETO

Da quest'anno l'abbonamento alle riviste Jackson offre una serie innegabile di vantaggi e servizi: anzitutto lo sconto eccezionale del 40% sul prezzo di copertina, pressochè doppio rispetto al passato, che Jackson ha voluto proporre ai lettori

SCONTO 40%

per celebrare il decimo anno di attività. Inoltre, abbonarsi a Jackson garantisce l'accesso a una rete multinazionale di informazioni, grazie al recente accordo azionario con la VNU Business Press Group, maggiore editore tecnico internazionale del settore. Ma c'è di più: **la Jackson Gold Card,** per l'identificazione immediata del codice abbonamento, sarà recapitata gratuitamente agli abbonati e permetterà al titolare di usufruire di molteplici servizi gratuiti quali: sconto del 20% fino al 28/2/1989 e del 10% dopo tale data, sul prezzo di copertina di libri e software Jackson, per acquisti effettuati direttamente dall'editore, oltre a una serie di sconti per acquisti vari presso librerie, computershop e altri esercizi convenzionati in tutta Italia.

In più, il titolare di **Jackson Gold Card** potrà ottenere sconti sui corsi di formazione della Jackson S.A.T.A., la scuola Jackson di Alte Tecnologie Applicate, oltre all'abbonamento gratuito a 6 numeri di uno (a scelta) dei tre settimanali Jackson: "E.O. News Settimanale di Elettronica", "Informatica Oggi Settimanale" o il nuovissimo "Meccanica Oggi", annunciato per l'inizio del 1989.

Infine, l'abbonato ha diritto all'invio personalizzato e riservato dei cataloghi libri e della nuova rivista **"Jackson Preview Magazine",** con l'annuncio di tutte le novità editoriali Jackson.

ABBONAMENTO JACKSON = FORTUNA STREPITOSA

AUT. D.M. 4/72621 DEL 14/11/88

Abbonarsi alle riviste Jackson significa leggere il meglio, risparmiando il 40%, in informatica, elettronica e nuove tecnologie, ma soprattutto partecipare al grande concorso Jackson riservato agli abbonati, con la possibilità di vincere premi favolosi.

REGOLAMENTO DEL CONCORSO

1 – Il Gruppo Editoriale Jackson S.p.A. promuove un concorso a premi in occasione della Campagna Abbonamenti 1988/1989.

2 – Per partecipare è sufficiente sottoscrivere, entro il 31.3.1989, un abbonamento a una delle 30 riviste Jackson.

3 – Sono previsti 10 favolosi premi da sorteggiare fra tutti gli abbonati.

4 – Primo premio: un viaggio per due persone in Estremo Oriente, che prevede: passaggi aerei Swissair, pernottamenti in Hong Kong, Bangkok e Singapore, presso gli hotel: Royal Orchid Sheraton e Sheraton Towers della catena Sheraton Hotel, nonchè escursioni in luogo nelle tre suddette località.
Gli altri nove premi consistono rispettivamente (in ordine di esposizione) in:
1 computer Amiga 2000 completo di unità centrale con 1 MB di memoria, dischetto da 3" 1/2, tastiera, mouse, sistema operativo e monitor a colori 1084.
1 personal computer PC 20 III SERIE completo di unità centrale con 640 KB di memoria, dischetto da 5" 1/4, hard disk da 20 MB, mouse 1352, sistema operativo MS-DOS 3.20 monitor monocromatico e tastiera.
1 personal computer PC1 completo di unità centrale con memoria 512 KB, dischetto da 5" 1/4, tastiera, monitor monocromatico, sistema operativo MS-DOS 3.20 e GW-Basic.
1 computer Amiga 500 con 512 KB Ram e 256 KB Rom di memoria, sistema operativo e monitor a colori 1084.
1 computer "nuovo C64" completo di manuali e sistema operativo.
Dal settimo al decimo premio incluso, n. 4 pacchetti "Commodore Software by CTO".

5 – Gli abbonati a più di una rivista avranno diritto, per l'estrazione, all'inserimento del proprio nominativo tante volte quante sono le testate sottoscritte.

6 – L'estrazione dei 10 premi in palio avverrà presso la sede del Gruppo Editoriale Jackson entro il 30.5.1989.

7 – L'elenco dei vincitori, ad estrazione avvenuta, pubblicato su almeno 10 delle riviste Jackson. La vincita inoltre sarà pubblicata con lettera raccomandata a ciascuno dei sorteggiati.

8 – I premi verranno messi a disposizione degli aventi diritto entro 30 giorni dalla data dell'estrazione, ad esclusione del primo premio, il quale dovrà essere effettuato, compatibilmente con la disponibilità dei posti, in un periodo da definirsi, entro il 31.12.1989.

9 – Le spese di vitto relative al viaggio, nonchè l'eventuale controllo di manutenzione extra garanzia per i personal computer Commodore, saranno a carico dei rispettivi vincitori.

10 – I dipendenti, i familiari, i collaboratori del Gruppo Editoriale Jackson sono esclusi dal concorso.

Sempre quest'anno, il concorso abbonamenti Jackson prevede un primo premio veramente eccezionale: la possibilità di esplorare il misterioso Estremo Oriente, in un viaggio che unisce il fascino di una tradizione millenaria ad uno sviluppo tecnologico senza precedenti.

Il viaggio, di oltre dieci giorni per due persone, è studiato nei minimi dettagli, per offrire il miglior comfort e le migliori ospitalità ed è garantito da tre leader di primissimo livello: Acentro Turismo di Milano, Swissair e Sheraton Hotels.

Non solo. Ad altri nove abbonati fortunati, il Gruppo Editoriale Jackson, in collaborazione con Commodore Computer e CTO, riserva altri premi eccezionali, dalla più completa gamma di computer di successo: un favoloso personal computer Amiga 2000, un Commodore PC20 III serie, un Commodore PC1, un Amiga 500 e un nuovo C64, in palio dal secondo al sesto estratto. Quattro pacchetti "Commodore Software by CTO" saranno inoltre sorteggiati dal settimo al decimo premio.

Partecipare al concorso è semplice: basta abbonarsi a una o più tra le riviste Jackson (chi si abbona a più riviste ha, naturalmente, più possibilità di vincita), utilizzando la speciale Cartolina/Questionario, già predisposta e affrancata, da compilare in ogni sua parte e restituire all'editore.

Affrettatevi! Abbonatevi per vincere!

Commodore
LEADER IN PERSONAL COMPUTER

GRUPPO EDITORIALE JACKSON
PRIMO NELLA BUSINESS-TO-BUSINESS COMMUNICATION

Abbonarsi è semplice: basta compilare in ogni sua voce la speciale Cartolina/Questionario già predisposta e affrancata e rispedirla all'editore.
Per il versamento dell'importo dell'abbonamento, utilizzate, preferibilmente l'apposito modulo di C.C.P. già predisposto e allegato alla rivista.

SPECIALE: PER CHI ACQUISTA LE RIVISTE JACKSON IN EDICOLA

Da quest'anno il Gruppo Editoriale Jackson ha predisposto uno **Speciale Servizio di Qualificazione Lettori e Abbonati,** che prevede l'assegnazione di una serie di dati relativi agli interessi specifici di ognuno, per poter offrire un servizio adeguato alle reali esperienze di aggiornamento del lettore.
Tutti i lettori interessati allo **Speciale Servizio Qualificazione Lettori,** e quindi anche i non abbonati, devono restituire, compilata nella parte **Qualificazione lettori,** la **Cartolina Questionario** già predisposta e affrancata.

Per chi la spedirà, il Gruppo Editoriale Jackson garantisce fin d'ora **GRATUITAMENTE:**

- Jackson Silver Card, che offre tutti i vantaggi della Gold Card, esclusi gli sconti sui libri riservati agli abbonati.

- Invio gratuito del **Catalogo Generale Libri Jackson.**
- Invio gratuito della **Jackson Preview Magazine.**
- **Abbonamento gratuito a sei numeri,** a scelta tra le seguenti riviste settimanali: E.O. News Settimanale - Informatica Oggi Settimanale - Meccanica Oggi (pubblicato da febbraio '89)

ABBONAMENTO JACKSON = RISPARMIO ECCEZIONALE

Area	Testate	Numeri Anno	Tariffa abbonam.	Tariffa intera
Elettronica e automazione	EO News Settimanale	40 + 6 omaggio	£. 59.500	£. 100.000
	Elettronica Oggi	20	£. 60.500	£. 100.000
	Automazione Oggi	20	£. 60.000	£. 100.000
	Meccanica Oggi	40 + 6 omaggio	£.59.000	£. 100.000
	Strumentazione e Misure Oggi	11	£ 39.000	£. 66.000
Informatica e Personal Computer	Informatica Oggi Settimanale	40 + 6 omaggio	£. 61.000	£. 100.000
	Informatica Oggi mese	11	£. 33.500	£. 55.000
	BIT (quindicinale da Gennaio)	20	£. 48.000	£. 80.000
	PC Magazine	11	£. 32.500	£. 55.000
	PC Floppy	11	£. 79.500	£. 132.000
	Computergrafica e applicazioni	11	£. 39.500	£. 66.000
	Trasmissione dati e Telec.	11	£. 34.000	£. 55.000
	Compuscuola	10	£. 24.500	£. 40.000
Tecnologie e mercati	WATT (quindicinale da Gennaio)	20	£. 36.500	£. 60.000
	LAB. NEWS	10	£. 30.000	£. 50.000
	Industria Oggi	11	£. 34.500	£. 55.000
	Media Production	11	£. 46.500	£. 77.000
	Strumenti musicali	11	£. 32.000	£. 55.000
Hobby e Home Computer	Fare Elettronica	12	£. 36.000	£. 60.000
	Amiga Magazine disk	11	£. 92.500	£. 154.000
	Amiga Transactor	6	£. 25.500	£. 42.000
	Commodore Professional 64/128 disk	11	£. 85.000	£. 143.000
	Commodore Professional 64/128 cass.	11	£. 59.500	£. 99.000
	Supercommodore 64/128 disk	11	£. 79.000	£. 132.000
	Supercommodore 64/128 cassetta	11	£. 49.500	£. 82.500
	Olivetti Prodest User	6	£. 18.000	£. 30.000
	PC Software	11	£. 66.000	£. 110.000
	PC Games 5 1/4"	11	£. 93.000	£. 154.000
	PC Games 3 ½	11	£. 99.500	£. 165.000
	3 ½ Software	11	£. 99.000	£. 165.000

Lo sconto del 40% è stato calcolato, in certi casi, arrotondando le cifre in modo da differenziare le tariffe di ciascuna rivista per esigenze di gestione.

GRUPPO EDITORIALE JACKSON

Defender of the crown

Quelli che vi proponiamo sono alcuni accorgimenti, che se utilizzati vi permetteranno di proseguire più agevolmente il gioco. Durante l'assedio al castello dovrete cercare di non sprecare tutte le vostre munizioni prima di aver aperto una breccia nelle mura del castello. Compito questo più facile a dirsi che a farsi!

Per incrementare la mira dei vostri lanci, ci sono alcuni accorgimenti che possono ritornare utili e non vi faranno sprecare troppi colpi. Il primo lancio va effettuato con la culla della catapulta posizionata in modo che siano esposti solamente cinque o sei pixel della corda. Ogni colpo seguente va effettuato in modo da visualizzare un pixel in più della corda. In questo modo riuscirete in breve ad aprirvi un varco nelle mura.

Un altro fatto degno di nota è che se il vostro maniero viene attaccato ed è sprovvisto di difese (truppe), il computer provvederà in modo automatico a porre nel castello un numero di uomini pari alle vostre compagnie campali. Per chiarire il concetto vi forniamo un esempio: nel vostro castello non ci sono truppe e avete 70 uomini nella vostra campagna militare quando il vostro maniero viene attaccato dalle forze avversarie. Il computer in questo caso provvederà a fornire il vostro castello di 70 uomini, mantenendo inalterato il numero di uomini presenti nella vostra campagnia militare. Se riuscite a sconfiggere i vostri avversari, verrete premiati, potrete cioè ritornare alla vostra campagna militare e aggiungere questi 70 uomini a quelli già presenti sul campo, raddoppiando in questo modo le vostre truppe d'attacco.

Sinbad and the throne of the falcon

Se non apprezzate il fatto di essere continuamente estromessi dal territorio nemico o uccisi ogni volta che affrontate un incontro, vi proponiamo un semplice accorgimento che vi farà risparmiare delle vite. Premendo due volte il pulsante sull'opzione posta in alto sulla sinistra dello schermo ci si immette nel GAME UTILITIES. Si scelga al suo interno l'opzione RESUME GAME che vi riporterà alla schermata della mappa del mondo. Ora potete continuare le vostre esplorazioni.

Nel caso siate pronti per l'incontro finale con il Principe Nero che si svolge nel castello, provate quanto sopra descritto e vi accorgerete che le vostre forze risulteranno ai massimi valori.

E' chiaro che questo fatto non renderà più facile la sconfittà del Black Prince, ma potrete comunque vivere un po' più a lungo e sfruttare quest'opportunità a vostro vantaggio.

Barbarian

Lavorare duramente per superare un livello ed essere poi distrutti da qualche strana creatura non è il massimo delle aspirazioni. La possibilità di evitare questi inconvenienti vi solleverà da future frustrazioni. Per ottenere questo non dovrete far altro che premere il pulsante sul gadget PICK UP un attimo prima che la creatura vi sovrasti, così disabiliterete il trap relativo alla collisione e i mostri vi attraverseranno senza crearvi alcun danno. Questo trucchetto è valido con tutti i mostri ed anche con i draghi. Ed è possibile, in questo modo, disattivare molte delle trappole presenti sul percorso.

Garrison

Vi proponiamo alcune osservazioni sul modo di rimanere vivi più a lungo in questa versione del più famoso Gauntlet presente nei Bar. Scegliete sempre tutti e cinque i personaggi. La forza iniziale dei personaggi è identica e non dipende dal numero scelto. Non cercate di raccogliere tutti forzieri, dal momento che ciò non incrementa il vostro punteggio, raccogliete solamente quello che vi è strettamente necessario. Anche se giocate da soli potete avvantaggiarvi della modalità due giocatori. Con un personaggio eseguite il lavoro e appena questi sta per uscire, prendete un nuovo personaggio e eliminate quello vecchio. Sparare ai vari mostri è sempre preferibile al combattimento corpo a corpo. Non preoccupatevi del tempo, la salute del vostro personaggio sarà danneggiata in misura minore nei combattimenti a distanza. I combattimenti ravvicinati richiedono sempre un maggior dispendio di energia e spesso il risultato non è nemmeno gratificante.

F/a - 18 interceptor

Ovvero, come volare con il più sofisticato velivolo del nostro tempo. Il programma che vi presentiamo è stato progettato da Robert Dinnerman, autore che si è interessato di prospettiva in tre dimensioni e illusione del volo fin dal periodo in cui lavorava presso la Motorola come disegnatore di hardware. Robert decise di provare la sua abilità in questo campo dopo aver visto un simulatore di volo prodotto da Evans e Sutherland. Giocando con Interceptor ci si accorge immediatamente e penso che sarete d'accordo con noi che abbia fatto un ottimo lavoro.

Il programma può essere utilizzato con i modelli AMIGA 500, 1000, 2000, ma prima di iniziare a giocare è bene che proteggiate il dischetto spostando l'apposita tacca presente sulla finestrella. Poi preparatevi una copia del dischetto, per coloro che avessero delle difficoltà nell'eseguire quest'operazione, facciano riferimento al manuale allegato all'Amiga su come si esegue la formattazione di un nuovo dischetto e su come si copiano i dischetti. E' possibile giocare sia utilizzando la tastiera che il joystick. Il programma richiede un minimo di 512K di memoria, perciò se il vostro computer ha solamente 512K dovrete sconnettere eventuali drive esterni. Per apprezzare al

massimo le capacità sonore e i colori si dovrebbe utilizzare un monitor RGB a colori e una macchina con un megabyte di memoria.

Il programma verificherà la vostra abilità nel combattimento aereo alla guida di un modernissimo jet da combattimento,

rasentando la baia per poi librarsi sulla penisola di San Francisco e effettuare delle manovre sull'Oceano Pacifico. Potrete scegliere tra due velivoli, il primo è un F/A-18 Hornet della Marina Militare mentre il secondo è un F-16 Fightin Falcon dell'aviazione militare.

Per iniziare seguite passo a passo le seguenti istruzioni, studiando la sezione di riferimento sul velivolo per acquistare confidenza coi comandi del joystick e della tastiera. L'abitacolo ed i controlli sono gli stessi per entrambi gli aerei. Dopo aver osservato il volo dimostrativo e aver eseguito un volo libero, vi eserciterete in manovre di addestramento.

Per iniziare:

1 - Accendete il computer e il monitor;

2 - Inserite il mouse in porta 1 e il joystick in porta 2;

3 - Inserite la vostra copia di lavoro.

Se utilizzate un Amiga 1000, utilizzate il KickStart 1.2, poi inserite la vostra copia quando vi verrà richiesto il Work Bench. Spostate il puntatore sull'icona Program Disk, premete per due volte il pulsante di sinistra del mouse, poi premete nuovamente due volte il pulsante sull'icona dell'aereo presente al centro dello schermo.

4 - Appaiono a questo punto i titoli e quando richiesto iscrivetevi al Full Tour of Duty, ora potete premere il tasto Return.

5 - All'apparire dello schermo Selezione delle Missioni, non

dovrete far altro che premere il numero dell'opzione con la quale intendete iniziare a giocare.

Quando il disco non è protetto in scrittura, apparirà una R

joystick e spiccate il volo. Durante la fase di volo spostate il joystick verso la direzione nella quale intendete muovere il velivolo. Il pulsante di fuoco del jo-

(Rookie). Un Rookie è un pilota che ha terminato positivamente le prime tre missioni. In questo caso vi verrà permesso di selezionare direttamente una serie di missioni più ampia.

Per iniziare a giocare dovrete immettere un codice alfanumerico di sicurezza dopo aver selezionato una delle opzioni di volo. Quando vi verrà richiesto utilizzerete il Flight Computer (accluso al software) allineando le quattro lettere e/o numeri, poi digitate il codice che verrà visualizzato nella finestra sopra l'ultimo numero o lettera della sequenza fornitavi.

Dopo aver immesso la corretta sequenza premete il tasto Return e otterrete la conferma a procedere. Se immettete dei codici inesatti avrete solamente due altre possibilità per correggere l'errore, in caso contrario dovrete caricare nuovamente il programma.

Potrete guidare il velivolo o per mezzo del joystick o dalla tastiera. Mantenete fermo il joystick mentre state rullando sulla pista per il decollo. Quando avete raggiunto la velocità di 200 nodi tirate a voi la leva del joystick attiverà l'arma scelta e potrete tentare di cogliere i vari obiettivi che vi si presenteranno.

E' importante che otteniate una buona familiarità con i vari controlli del velivolo. Le caratteristiche principali dell'abitacolo sono il sistema HUD, i sistemi radar e delle armi a disposizione, l'indicatore di carburante, della velocità e l'altimetro. L'HUD, che compare al di sopra del pannello di controllo, comprende un indicatore del bersaglio, un reticolo di puntamento, un indicatore della scelta delle armi, un G-meter (misurazione di carico) e altri congegni ancora. Mentre esaminate la seguente lista di controllo riferitevi alle illustrazioni del pannello di controllo dell'abitacolo che sono accluse al software:

F1 - F10 : Spinta. Controllo della valvola con incrementi del 10%. Se premete due volte di seguito il tasto F1 spegnerete i motori se siete sul valore del 10%. Se la valvola è al 100%, premendo F10 accenderete l'afterburner.

= Key : Spinta. Incrementa la valvola di un passo (unità).

- Key : Spinta. Decrementa la valvola di un passo (unità).

Backspace : Freni. Attiva i freni sia durante il volo che durante la fase di atterraggio. L'indicatore dei freni sulla parte sinistra del pannello di controllo si accende quando vengono attivati.

, : Controllo timone. Left rudder, vira a sinistra.

. : Controllo timone. Right rudder, vira a destra.

Cursor Keys : Movimento aereo. La freccia verso l'alto abbassa il muso dell'aereo, freccia verso il basso e l'aereo si innalza, freccia verso sinistra e l'aereo ruota a sinistra, freccia verso destra e l'aereo ruota verso destra.

G : Gear. Alza o abbassa il carrello d'atterraggio.

A : Gancio d'arresto. Estende il gancio d'arresto per F/A 18 per l'atterraggio sulla portaerei.

H : HUD. Accende o spegne i display dell'HUD. Rimane comunque sempre attivo il display della scelta delle armi.

J : ECM. Accende o spegne il contatore di misura elettronico, che serve per disturbare la ricezione nemica. L'ECM si trova sulla parte destra del pannello.

C : Chaff. Dispositivo di difesa che genera un segnale di disturbo confondendo in questo modo il sistema elettronico di sorveglianza degli aerei nemici ed i missili a puntamento radar.

F : Flare. Dispositivo che invia dei razzi IR atti a sviare i missili IR guidati contro il vostro aereo.

M : Mappa. Visualizza mappa dello spazio aereo e la vostra posizione sul HSI.

R : Range. Controlla la risoluzione del radar (2, 10 o 40 miglia).

T : Selezione bersaglio. Premendo ripetutamente il tasto T si possono visualizzare i vari bersagli a disposizione.

RETURN : Scelta armi. Se premuto ripetutamente vengono visualizzate le varie armi a disposizione. L'arma scelta viene evidenziata con una barra di colore rosso nell'apposita window riservata alle armi nel pannello di controllo.

SPACE-BAR : Fuoco. Fa fuoco verso l'obiettivo prescelto.

SHIFT F : Attiva dei mezzi di salvataggio in alcune missioni.

SHIFT E : Espulsione. Da premersi quando il velivolo è condannato.

P : Pausa. Concede una pausa durante il gioco; si prema nuovamente lo stesso tasto per riattivare l'azione.

ESC : Reset Missione. Vi riporta all'inizio dell'attuale missione.

SHIFT ESC : Uscita dalla missione attuale. Si ritorna allo schermo di selezione delle mis-

sioni.

ENTER : Cambia il punto di osservazione da una prospettiva in prima persona (dall'interno dell'abitacolo) ad una prospetiva in terza persona (all'esterno del velivolo). Per rientrare nell'abitacolo si prema nuovamente il tasto ENTER.

SHIFT ENTER : Vi conduce alla torre di controllo più vicina alla vostra posizione di partenza.

[o] : Zoom. La parentesi di sinistra allontana, quella di destra vi avvicina.

Quando siete all'interno dell'abitacolo saranno disponibili i seguenti tasti del tastierino numerico:

4 : Sposta la testa a sinistra.

6 : Sposta la testa a destra.

5 : Fissa la testa nella posizione attuale.

2 : Muove la testa per vedere ciò che c'è dietro.

1 : Vista verso il basso.

7 : Vista verso l'alto.

8 : Vi riporta alla vista frontale.

I programma vi riporta in modo automatico alla vista frontale dopo che saranno trascorsi 4 secondi. Se premete il tasto ENTER rimarrà abilitato il tastierino numerico e varranno le stesse istruzioni appena lette e saranno attivati anche i tasti 3 e 9. Il tasto 5 permette la visione dell'aereo direttamente dall'alto; il punto decimale provoca la visione dell'aereo dal basso.

L'ECM vi sarà di notevole aiuto per confondere i radar nemici evitando di essere rilevati, impedendo così gli attacchi avversari. L'ECM può essere utilizzato sia in modo attivo confondendo i radar elettronici nemici che in modo passivo sperdendo nel cielo delle false tracce composte da piccole lamine o fili metallici che servono da segnali di richiamo e confondono i radar. Queste tracce confonderanno i rilevamenti dell'avversario eseguendo così un'ottima manovra diversiva. L'indicatore dell'ECM è posto sulla destra del pannello all'interno dell'abitacolo. Quando è attivato viene evidenziato con un colore giallo. L'utilizzo dell'ECM riduce l'efficacia dei missili avversari ma fornisce anche una segnalazione luminosa della vostra posizione.

Ogni velivolo ha a disposizione dei missili a corto e medio raggio e un cannone a breve raggio. L'AIM-120A AMRAAM è il missile a medio raggio a disposizione su ogni velivolo. Il raggio d'azione di quest'arma ricopre oltre 30 miglia e un sistema di guida mantiene la sua corsa inerziale in perfetta rotta di collisione con il bersaglio.

Il missile a corto raggio disponibile sul F/A 18 è il AIM-9L Sidewinder. Ottimo per le distanze non superiori alle 11 miglia. Quest'arma raggiunge la velocità di Mach 3. Il missile può essere sganciato quando si trova ad una distanza di 1000 piedi.

Il cannoncino da 20 mm. per le distanze ravvicinate è un GE-M61, quest'arma dimostrerà la sua validità non appena vi troverete in situazioni di pericolo ravvicinato; la sua capacità di fuoco è davvero impressionante riesce a sparare 570 colpi alla velocità di 6000 colpi al minuto.

Se a questo punto avete completato l'analisi di quanto fin qui illustrato possiamo proseguire con l'analisi del volo. Il punto da cui inizia l'avventura è sempre la schermata relativa alla scelta della missione, dove potrete scegliere la missione che intendete compiere. E' consigliabile scegliere la prima volta il volo dimostrativo in modo da familiarizzarsi con i comandi base e memorizzare dei punti di riferimento relativi alla baia di San Francisco. Poi potrete scegliere per i primi voli l'opzione Free Flight in modo da prendere dimestichezza con il velivolo. Ora passate a Training: Demo of Maneuvers in modo da apprendere i rudimenti delle manovre relative al combattimento. Per allenarsi con questo tipo di manovre si scelga l'opzione Training: Practice Maneuvers. La descrizione di ognuna delle manovre da effettuarsi in questa fase si trova nel Training: Demo of Maneuvers. Con Qualification si accede alla fase vera e propria del gioco scegliendo una delle sei missioni proposte.

Se scegliete l'opzione dimostrativa alla schermata selezione della missione premete il tasto con il numero 1. In questa fase non potrete controllare in alcun modo il velivolo. Il demo vi offre un assaggio di quello che vi aspetterà con il volo vero e proprio visualizzando anche l'atterraggio alla base aereonavale di Alameda. Il demo continuerà finchè non premete i tasti SHIFT ESC. Se siete pronti passate al modo Free Flight.

In questa modalità non dovrete affrontare nessun nemico, vi viene offerta l'opportunità di familiarizzare con l'aereo, ed è in questo modo che diverrete in breve tempo un esperto pilota da combattimento. Dopo aver scelto Free Flight immettete il codice di controllo, poi scegliete la locazione di partenza tra quelle che vi vengono offerte: 1 - San Francisco International; 2 - Alameda Naval Air Station; 3 - Moffett Field Naval Air Station; 4 - Aircraft Carrier off the Coast of

San Francisco. In questa fase è permessa anche la scelta dell'aereo che si intende pilotare (F/A-18 Hornet o F-16 Falcon). Verificate il pannello di controllo e quando tutti i comandi sono sulla posizione di GO premete il tasto F10. Quando la velocità raggiunge i 200 nodi tirate a voi il joystick per decollare.

In questa fase non dovrete affrontare alcun nemico nè ci saranno degli obiettivi da distruggere. Potrete però esplorare le possibilità del velivolo e familiarizzare con le capacità peculiari dell'aereo. L'accensione dei retrorazzi per il volo supersonico, le cabrate e gli atterraggi vi introdurranno ugualmente in quel clima tipico che è proprio del pilotare. Se malauguratamente vi succede di distruggere l'aereo sarete rinviati alla posizione di partenza. Se questo inconveniente accade durante una missione, ciò significherà che non otterrete i crediti prefissati per coloro che conducono a termine una missione.

Nella fase di Training: Demo of Maneuvers voi ricoprirete il ruolo del passeggero dal momento che sarà l'istruttore di volo a eseguire le sequenze necessarie alle manovre collegate al volo. Dopo aver scelto il Training Demo dalla schermata per la selezione della missione dovrete immettere un codice di sicurezza a questo punto potrete osservare dall'interno dell'aereo alla prima delle sette manovre. Il pilota non supererà in alcun momento l'80% delle possibilità offerte dal velivolo. Volerà ad oltre 8000 piedi dalla superficie terrestre e raggiungerà una velocità massima di 650 nodi.

Dal momento che sarete un semplice passeggero non avrete nessuna possibilità di influire sull'andamento del volo. Premendo Return potrete però interrompere una sequenza o portarvi ad una sequenza successiva. Una volta completata la sequenza delle operazioni questa verrà ripetuta automaticamente finchè non decidiate di uscire dal demo (SHIFT ESC). Con quest'opzione potrete apprendere teoricamente alcune tecniche che in seguito vi saranno utili.

Nella fase di Training: Practice Maneuvers le operazioni saranno identiche a quelle della fase precedente, perciò eseguitele nell'identico ordine. Immediatamente dopo aver scelto questa modalità immettete il codice di controllo, poi leggete le istruzioni che verranno visualizzate sullo schermo. Siete sulla pista di decollo e davanti a voi c'è il velivolo dell'istruttore che attende che decolliate. Accelerate fino all'80% della spinta di cui il velivolo può disporre, quando raggiungete i 200 nodi potete prendere il volo. Ora osservate l'istruttore ed eseguite tutte le manovre che egli compie. Se vi succede di perdere il contatto con l'istruttore vi verrà in aiuto il tasto Return che vi permette di riposizionarvi sulla sua scia, fate in modo di rimanere sempre dietro all'istruttore senza sorpassarlo.

Cercate di utilizzare quello che avete fin qui appreso e seguite l'istruttore nelle sue acrobazie. Per evitare manovre particolarmente difficili premete il tasto Return e se volete del fumo tra il vostro aereo e quello dell'istruttore premete il tasto S, in questo modo l'aereo dell'istruttore lascierà dietro di se una scia di fumo. Anche in questo caso le manovre verranno ripetute all'infinito a meno che non premiate i tasti SHIFT ESC che vi riporteranno alla schermata dedicata alla scelta della missione.

Vi descriviamo brevemente le sette manovre che sono presenti nel Training:

Aileron Roll viene eseguito muovendo il joystick su un lato. La punta dell'aereo decriverà un cerchio. E' una tecnica che in combattimento vi sarà molto utile dal momento che vi permette di passare rapidamente dalle risalite alle picchiate.

Break Turn vi servirà per interrompere l'avvicinamento di un aereo avversario uscendo dal suo inquadramento.

Inside Loop, una manovra classica che viene effettuata tirando a se il joystick vi permetterà di mettervi in coda ad un aereo che vi segue da vicino.

Con Split S forzerete l'attacco del vostro avversario facendogli, si spera, fallire il bersaglio. In questa manovra si vola rovesciati poi si scende verticalmente in picchiata muovendosi in direzione opposta a quella dell'avversario.

Vertical Half-Loop, è un semiloop che vi farà volare in senso contrario dopo che avrete completato la manovra ruotando su se stesso l'aereo di 180 gradi. Questa manovra vi permette di guadagnare quota invertendo la direzione.

Barrel Roll, dopo aver effettuato una cabrata si volerà nella direzione opposta. Anche se la perdita di velocità è notevole vi sarà ugualmente utile se siete seguiti da aerei che si trovano comunque a un'altezza superiore alla vostra.

Inverted Flight è un elemento

base di varie figure aeree.

Quando vi sentite sufficientemente pronti sceglierete l'opzione Qualificazione sullo schermo prefigurato per la scelta delle missioni. Udrete il rombo dei motori dell'F/A-18 e potrete osservare il ponte dell'Enterprise che starà tranquillamente navigando sull'oceano. Dopo aver immesso il consueto codice di controllo seguite le istruzioni per le qualificazioni che appariranno sullo schermo. In questa fase dovrete innalzarvi in volo, volare e atterrare sul ponte della portaerei. Queste operazioni vi prepareranno ad eventuali attacchi futuri. Una volta che vi sarete qualificati potrete accedere alla fase di effettivi. Questo avanzamento vi verrà notificato con un messaggio sullo schermo.

La pressione del tasto Escape vi riporta sul ponte di volo e la pressione dei tasti SHIFT ESC vi riporterà al quadro della scelta della missione.

Nello scegliere Next Active Advanced Mission (Missioni

Avanzate) accetterete di effettuare delle missioni di combattimento. Dopo aver selezionato questa opzione e inserito il codice di controllo, eseguirete la prima di una serie di missioni (4). Per ognuna dovrete svolgere un compito specifico. Per riportare il sucesso in questa fase dovrete essere degli esperti nel combattimento, saper sorprendere il nemico senza farvi scoprire, portare degli attacchi, scegliere e preparare con cura i vari sistemi offensivi.

Per aiutarvi, nell'angolo in alto a destra del pannello di controllo, verranno emesse delle luci di diverso colore e un messaggio apparirà sullo schermo:

VERDE = aereo amico in zona;
ROSSO = nemico in zona;
ROSSO = allarme di stallo;
BIANCO = missile RH in avvicinamento;
BIANCO = missile IR in avvicinamento.

Visual Confirmation è la prima missone di combattimento. Decollerete dall'aereoporto di San Francisco e dovrete cercare e identificare un aereo. L'aereo, ovviamente, potrà essere nemico o amico, esiste perciò il pericolo che possiate essere attaccati. I vari sistemi offensivi sono operativi e se sarà necessario attivateli. Se qualcuno vi spara rispondete al fuoco; a missione conclusa un messaggio apparirà sullo schermo e potrete rientrare alla base.

Emergency Defence Operation sarà la seconda missione, nella quale siete chiamati ad affrontare una operazione di difesa di emergenza. Ora dovrete decollare dal ponte di una portaerei ancorata al largo del-

le coste di San Francisco. Il vostro compito è quello di tenere occupato l'aereo nemico concedendo all'aereo del presidente il tempo necessario per atterrare sano e salvo. Anche in questo caso un messaggio sullo schermo vi informerà se la missione ha avuto successo. Questa sfida richiede una notevole dose di abilità dal momento che dovrete conquistare una posizione di attacco prima, sviare e annientare l'avversario poi.

Con Interceptor Stolen Aircraft vi troverete nella situazione in cui due piloti della Air Force hanno disertato e sono fuggiti con i loro aerei, aiutati nella loro fuga da una scorta non identificata. Dovrete decollare dalla portaerei, rintracciare i disertori e la loro scorta. Volate davanti ai fuggiaschi per informarli che sono sotto controllo. Se non oppongono alcuna resistenza dovrete semplicemente scortarli alla portaerei, in caso contrario dovrete comunque impedirne la fuga.

Search and Rescue Mission vi coinvolgerà in un'operazione di ricerca e di soccorso. Dovrete localizzare un vostro collega abbattuto dal nemico. Dopo aver decollato dall'Enterprise cercate di individuare il disperso utilizzando il radar e la vostra ottima vista. Può succedere che il nemico cerchi di contrastarvi. Una volta individuato il pilota, sganciate il rescue pod nelle sue vicinanze. Se l'azione ha successo verrà confermata da un messaggio sul video. Ora, stanchi ma soddisfatti, potrete ritornare a bordo della portaerei.

Dopo aver concluso Next Active Advanced Mission potrete accedere a Selectable Mission che vi permetterà di scegliere una delle missioni appena portate a termine, più un'altra ancora. In questa fase, ogni volta che completate una missione vi verrà fornito un nuovo scenario. Quando partite per una missione, tutte le armi e gli strumenti sono operativi. Selectable Mission vi servirà per impratichirvi con lo scenario di battaglia.

Saranno a disposizione anche altre missioni selezionabili e se terminate con successo una missione attiva potrete ripeterla.

Il Current Flight Log vi permette di analizzare i vostri pro-

gressi al momento della scelta di un Tour of Duty.

Scegliendo quest'opzione dallo schermo per la selezione delle missioni vi troverete una lista delle seguenti missioni:

1 - Totale delle missioni affrontate;
2 - Missioni terminate con successo;
3 - Totale delle armi azionate e i colpi andati a segno;
4 - Tempo di volo accumulato;
5 - Numero complessivo degli incidenti, le espulsioni non saranno conteggiate;
6 - Colpi ricevuti dai missili nemici.

Il Flight Log viene aggiornato dopo ogni missione. Per registrarlo su dischetto premete il tasto 1. Ogni missione completata aumenterà il vostro credito. Ad esempio, se avete completato le prime tre missioni, potrete iniziare la terza non appena lo desiderate. Se volete cancellare il Flight Log con le statistiche, premete i tasti SHIFT 2. Per uscire dal Flight Log si dovrà ritornare allo schermo di scelta delle missioni e premere il tasto ESC.

Per concludere vorremmo fornirvi alcuni consigli. Apprendere il lavoro dei combattenti è l'elemento principe nell'esperienza di un ottimo pilota. Si debbono sapere le posizioni dei controlli anche a occhi chiusi. L'F/A-18 e l'F-16 sono il meglio che la Marina e l'Aereonautica possano mettervi a disposizione. Nelle vostre mani si trasformano in strumenti di precisione e l'esperto pilota si distingue per essere un vero acrobata del cielo. L'arte del volo di combattimento è la somma di diverse abilità. L'esperienza è la migliore fonte di apprendimento per ogni pilota e non c'è niente che possa sostituire le lunghe ore di addestramento nelle manovre di volo da cui si traggono le norme per l'attacco e la capacità di concluderlo vittoriosamente. Le qualità richieste sono: il coraggio, la determinazione, la fiducia nelle proprie capacità e una notevole dose di sangue freddo.

Un bravo pilota affina le tecniche del volo durante gli addestramenti. Questo concetto implica addestramenti graduali, partendo da Free Flight fino ad arrivare a Next Active Advanced Mission. Regole a cui si deve sottostare sono: la scoperta del nemico, la conquista di una posizione d'attacco vantaggiosa, un rapido avvicinamento e un preciso utilizzo delle armi. Un pilota esperto ha nel suo bagaglio un discreto repertorio di manovre. Un'accurata preparazione vi renderà sicuri anche nei conflitti aerei più difficili.

Arkanoid

Il capitano dell'astronave galattica Arkanoid ricevette la notizia come si confaceva ad un alto ufficiale della Federazione Terrestre. Ascoltò, interrompendo soltanto per porre alcune domande e non appena l'ufficiale tecnico ebbe lasciato la cabina si versò un whisky. Il suo sguardo errò, senza percepirli, sui visori.

Aveva viaggiato e aveva visto molte cose ma era ancora abbastanza giovane da sentirsi raggelare il sangue nell'udire la sua sentenza di morte. Lo schermo mostrava una incredibile moltitudine di stelle e soltanto un astronauta avrebbe potuto identificarne alcune. Il capitano cercò oltre la Via Lattea, fin quando non identificò la Stella Polare. Provò un certo conforto sapendo che stava guardando verso la più vicina base della Federazione Terrestre, specialmente ora, convinto com'era che non vi sarebbe atterrato mai più.

Intorno a lui la nave ronzava a una pseudo velocità che lasciava la luce molto indietro, e che tuttavia era sempre troppo lenta per salvarli. Il dovere di un capitano era quello di pensare, prima di tutto, agli altri. Perse un attimo a controllare il suo aspetto; il morale era importante, ora più che mai. Invece della solita tuta grigia di bordo, preferì indossare l'uniforme completa. Scese nella stiva principale che occupava oltre la metà del volume dell'astronave. Una serie infinita di cubicoli era allineata sui vari ponti della stiva, piastre fluorescenti movibili erano poste sul soffitto e il pavimento era stato rivestito di plastica elasticizzata, chimicamente stabile.

Gli umani in animazione sospesa potevano essere controllati dal lato frontale dei cubicoli; questo lato costituito da plastica espansa trasparente era l'unico modo per poter mantenere un contatto, puramente visivo, con dei compagni che sarebbero stati rianimati appena alla fine del viaggio interstellare e cioè tra qualche centinaio d'anni. Nella stiva erano raccolti 4,5 milioni di esseri

umani; erano gli unici sopravvissuti al brutale attacco alieno alla Terra ed erano anche l'unica speranza che rimaneva al genere umano per non essere cancellato dall'Universo.

Ad un tratto una serie di flash multicolori avvolsero l'astrona-

non siano tutti distrutti.

Il game ha elaborato questo concetto sviluppandolo in modo sorprendente. Alla fine dei trentadue livelli del gioco viene aggiunto uno schermo con un episodio finale. L'ottima grafica è accompagnata da un'animazione fluida. Lo schermo è colmo di mattoni multicolori, che se colpiti determinano l'acquisizione di un punteggio diverso, e che a seconda delle schermate formano una configurazione d'insieme totalmente differente. Il gioco classico è formato da varie serie di questi mattoni, ma questo game è fornito per ogni schermata di diverse configurazioni. Ma ciò che distingue questo game è la presenza ad ogni schermata di una serie di opzioni disponibili. In primo luogo sono presenti mattoni con diversi colori. Quelli di color argento devono essere colpiti varie volte prima di venire distrutti. Il numero dei colpi necessari per distruggerli aumenta con il passare degli schermi. Quelli dorati sono indistruttibili, perciò non sprecate il vostro tempo cercando di romperli. Altri, se spezzati, liberano delle capsule. Queste capsule sono di sette tipi diversi e se catturate vi forniranno di speciali caratteristiche. Potrete rallentare la velocità della pallina, cosa questa estremamente utile ai livelli più difficili nei quali la pallina raggiunge una velocità incredibile.

Un'altra capsula vi permetterà di allargare del doppio la "barretta" con la quale colpite la pallina. Altre vi faranno superare immediatamente la schermata e guadagnare una barretta extra. Se catturate la Disruption, appena colpirete la pallina questa verrà triplicata. Anche se diventa difficile seguire contemporaneamente la traiettoria di tre palline, questa possibilità vi tornerà utile per colpire dei mattoni posti in posizioni scomode. La Laser, se catturata, vi permette di trasformare la vostra barretta in un carro armato fornito di laser. Premendo il pulsante del mouse attiverete il fuoco contro i mattoni. Quest'opzione risulta utilissima quando vi trovate con delle schermate zeppe di mattoni argentati.

Se non siete ancora soddisfatti, e pensate che gli oggetti da seguire non sono poi molti, i disegnatori hanno pensato di allietarvi con continue invasioni di Konerds, Pyradoks, Trispheres e Opopos. Anche se facilmente distruttibili devieranno i vostri sforzi dall'obiettivo principale che è quello di distruggere più mattoni possibili.

Quest'insieme di opzioni possono generare indubbiamente molta confusione. Ricordatevi però che la capsula che catturate trasforma la vostra barretta. Facciamo un esempio: se avete il Laser e si avvicina una capsula Slow (lenta) appena questa viene catturata, l'opzione Laser viene annullata e viene attivata l'opzione Slow. Perciò dovrete cercare di pianificare le vostre scelte a seconda delle vostre necessità.

Il gioco è veramente ben fatto e una volta completata la prima partita, desidererete riprovare immediatamente il gioco per poter vedere gli schermi successivi.

L'azione è sempre brillante e la grafica superba. Se siete alla ricerca di un arcade game che non vi annoi anche se viene ripetuto decine di volte, Arkanoid è il vostro game.

CLASSIFICHE

APPLICATIVI

1	CAO 3D	Infogrames
2	MATH ANIMATION	Progressive Peripherials
3	KINDWORDS	The Disco Company
4	DELUXE PAINT II	Electronic Arts
5	DELUXE VIDEO	" "
6	DELUXE PRINT	"
7	INTROCAD	Progressive Peripherials

GIOCHI

1	SPEEDBALL	MirrorSoft
2	THUNDERBLADE	US Gold
3	OPERATION WOLF	Ocean
4	TV SPORT FOOTBALL	MirrorSoft
5	CALIFORNIA GAMES	US Gold
6	F-16 FALCON	MirrorSoft
7	DUNGEON MASTER	MirrorSoft
8	ELITE	Firebird
9	INTERNATIONAL SOCCER	Microdeal
10	HEROES OF THE LANCE	US Gold

NON PERDETE D'OCCHIO
LA VOSTRA EDICOLA

È IN ARRIVO...

CON IL 1° NUMERO
DI AMIGA MAGAZINE GAMES

PRENOTATELO!!!

GRUPPO EDITORIALE
JACKSON
AREA CONSUMER

di Paolo Russo

Ebbene sì, è accaduto ciò che molti temevano: in redazione sono giunte molte lettere esprimenti un certo disappunto per l'eccessiva difficoltà del corso di Amiga-Basic. Quando si decide di creare una rubrica fissa ci si chiede anzitutto quale ne sia il target, cioè il pubblico al quale è destinata; dopo una certa riflessione si era giunti alla conclusione che non molti acquistano un Amiga come loro primo computer e che di conseguenza il lettore medio dovrebbe possedere già una certa dimestichezza con il Basic, magari di un'altra macchina. Considerando anche il fatto che esistono sul mercato innumerevoli testi che insegnano il Basic generico (e nessuno che spieghi quello dell'Amiga, il cui manuale è, tra l'altro, assai carente) la scelta più logica sembrava consistere in un corso che illustrasse come passare da un qualunque Basic a quello, molto più complesso e articolato, dell'Amiga. Dietrofront dunque, si riparte da zero; ma come non deludere il pubblico (forse abbondante) del target iniziale, che vuole passare subito al sodo senza tergiversare troppo su argomenti già noti? Da questo momento ogni puntata sarà divisa in due parti, la prima delle quali destinata al neofita, la seconda all'"iniziato". Fateci sapere cosa ne pensate, le buone proposte non vengono mai destinate al Trashcan.

Partiamo da zero: i dischi di sistema

Non avete mai programmato in Basic in vita vostra? Bene, anzi male, ma non temete: siamo qui per questo. Dunque, cominciamo dall'inizio. Procuratevi qualche disco vuoto, avendo cura di richiedere il tipo da tre pollici e mezzo, doppia faccia, doppia densità (3.5'' DS/DD, Double Side / Double Density). A costo di annoiare qualcuno, descriveremo nei minimi dettagli anche le operazioni più usuali, perfino quelle che non hanno strettamente a che fare con il Basic, onde non ripetere l'errore iniziale. Avete i dischi con voi? Bene, procediamo.

I due dischi che vi sono stati sicuramente consegnati assieme al computer si chiamano Workbench ed Extras; i possessori di un Amiga 1000 avranno anche il Kickstart. Occorre creare subito una copia (almeno) di sicurezza di ognuno di questi dischi; lo Workbench (con il Kickstart, se c'è) è indispensabile per lavorare con l'Amiga ed il disco degli Extras contiene il Basic. Copiare un disco è semplice. Si accende l'Amiga, si inserisce nel drive il Kickstart (solo sul 1000) e poi lo Workbench; si attende poi pazientemente per una quarantina di secondi (almeno con la release 1.2) che il computer esca dalla fase di bootstrap (questo è il nome che viene dato alla fase iniziale di funzionamento di ogni computer). A questo punto potremo contemplare lo schermo dello Workbench.

Inseriamo in un qualunque drive il disco che vogliamo copiare e aspettiamo che sullo schermo compaia la relativa icona; a questo punto muoviamo la freccina (che si chiama pointer) su tale icona con il mouse e premiamo una volta il tasto di sinistra dello stesso. L'icona cambierà colore; l'abbiamo appena selez*ionata*. Non per nulla il tasto di sinistra prende il nome di select button, in contrapposizione a quello di destra chiamato menu button (sembra che la parola menu, usata come termine tecnico, non voglia l'accento).

Teniamo costantemente premuto il ta-

sto di destra e vedremo apparire sulla barra superiore dello schermo i nomi dei tre menu dello Workbench: Workbench, Disk e Special. Tenendo sempre premuto il pulsante spostiamo il pointer sul primo di essi e vedremo apparire le sue opzioni: Open, Close, Duplicate, Rename, Info e Discard. Muoviamo il pointer su Duplicate e finalmente rilasciamo il tasto di destra; abbiamo appena ordinato allo Workbench di creare una copia dell'oggetto selezionato, cioè il disco. Se non accade nulla, qualcosa è andato storto; in tal caso ripetete tutte le operazioni con più attenzione a partire dalla selezione del disco da copiare in poi.

Se non avete commesso errori appariranno dei messaggi in finestrelle che sono chiamate requester perché si aspettano una risposta dall'utente; nella parte bassa del requester sono elencate le possibili risposte e spesso ne dovete selezionare una andandoci sopra con il pointer e premendo il tasto di sinistra del mouse (cioè click*andoci* sopra, come spesso si suol dire). In alcuni casi ciò non è necessario; se vi viene chiesto di inserire un certo disco in un drive (insert Caio disk in any drive), dopo aver adempiuto alla richiesta non è (di norma) necessario clickare su alcunchè perché il sistema si accorge da solo che il disco desiderato è stato reso accessibile.

Per motivi sconosciuti (un bug, probabilmente) il sistema talvolta non se ne accorge; in questo caso occorre clickare su Retry. Talvolta appare un requester che vi chiede di togliere la protezione dalla scrittura (disk Tizio is write protected) ed anche in questo caso non dovrebbe essere necessario clickare nulla. Quasi tutti i requester prevedono Cancel tra le possibili risposte; tale opzione provoca l'interruzione dell'operazione in corso.

Durante la copia del disco appariranno, come già accennato, dei requester; fate ciò che essi vi chiedono e poi clickate su Continue se necessario. Alla fine apparirà l'icona del nuovo disco. Fate una copia di tutti i dischi che prevedete di usare, riponete poi l'originale in un cassetto e lavorate sempre sulla copia.

Se disponete di due drive il processo di duplicazione è molto più semplice. Inserite in un drive il disco da copiare e nell'altro quello vuoto che accoglierà la copia; a questo punto dovreste vedere nello schermo le icone di entrambi. I dischi vergini (o comunque mai usati prima da un Amiga) appaiono con il nome DFn:BAD. Centrate il pointer sul disco da copiare, premete il select button e mantenetelo premuto; muovete il pointer sull'altro disco e rilasciate il tasto. Avrete notato che l'icona del primo disco si è mossa con il pointer; avete appena eseguito un dragging (trascinamento). Il sistema ha capito che il contenuto del primo disco deve essere ricopiato nel secondo. Un consiglio: prima di copiare un disco proteggete sempre l'originale aprendo l'apposita finestrella, onde evitare di scrivere sul disco sbagliato; ne vale senz'altro la pena.

L'ambiente Basic

Accendete l'Amiga e inserite il disco dello Workbench nel drive interno (dopo il Kickstart, se necessario); al termine del bootstrap inserite in un qualunque drive il disco degli Extras (la copia, s'intende...) e clickate sulla sua icona due volte in rapida successione (biclickaggio). Si aprirà una finestra contenente altre icone; alcune di queste rappresentano cassetti (drawer) che si possono aprire a loro volta con la medesima operazione. Una delle icone più grandi è quella dell'interprete AmigaBasic; biclickateci sopra e in pochi secondi sarete in ambiente Basic.

Le finestre associate al Basic sono due: la più grande, sulla cui barra superiore spicca la scritta BASIC, si chiama output window mentre l'altra, con la scritta LIST, è la list window (per l'appunto). Quest'ultima copre parzialmente la metà destra dell'output window sottostante. E' adesso opportuno aprire una parentesi per illustrare diversi concetti di validità generale su Intuition e lo Workbench, la cui completa padronanza è indispensabile a chiunque voglia usare l'Amiga, non solo ai programmatori Basic; buona parte dei lettori avrà già acquisito una certa confidenza con Intuition, ma questo corso (come spiegato nell'introduzione) parte veramente da zero.

Una generica finestra può possedere i seguenti elementi, chiamati genericamente gadget:

1) Una barra superiore con il nome (o titolo che dir si voglia) della finestra (title bar o drag bar). La barra può essere usata per spostare la finestra in un altra zona dello schermo; basta puntarci sopra il pointer e poi tenere premuto il select button muovendo nel contempo il mouse (dragging).

2) Due quadratini concentrici con un punto al centro, il tutto nell'angolo in alto a sinistra della finestra (close gadget). Clickandovi sopra la finestra viene chiusa, ovvero eliminata.
3) Due rettangoli con alcuni simboli all'interno, nell'angolo in alto a destra (back gadget e front gadget). Clickando sul primo dei due la finestra viene spinta dietro alle altre, mentre il secondo ha la funzione opposta.
4) Un rettangolo contenente una specie di otto obliquo, nell'angolo in basso a destra (sizing gadget). Eseguendo su di esso l'ormai consueto dragging possiamo cambiare le dimensioni della finestra.

Quelli sopra elencati sono i cosiddetti gadget di sistema, ossia quelli presenti nella stessa forma in quasi tutte le finestre. Possono esservene altri, come le scroll bar (rettangoli vuoti allungati con un rettangolo pieno al loro interno), utili per visualizzare un po' alla volta il contenuto delle finestre e presenti ad esempio in quelle dello Workbench. Le finestre del Basic possiedono solo i gadget di sistema.

Tra tutte le finestre aperte nel sistema una sola è quella corrente, quella cioè con la quale si sta lavorando in quel momento; distinguerla dalle altre non è difficile, essendo l'unica la cui barra superiore è ben nitida. Per rendere corrente una finestra basta clickare in un qualunque punto al suo interno. Ad esempio, quando si entra in ambiente Basic, la finestra corrente è inizialmente quella di list e qualunque cosa digitiate alla tastiera verrà dirottato al suo interno; volendo scrivere nell'output window bisogna prima clickarci dentro.

I menu scendono sempre dalla parte più alta dello schermo, ma sono in realtà associati alle finestre. Per la precisione, i menu che appaiono alla pressione del tasto di destra appartengono alla finestra corrente. Se volete convincervene afferrate l'output window per la sua drag bar e spingetela un po' in basso, in modo da scoprire lo Workbench sottostante. Se clickate in un punto dello stesso (che è in realtà una finestra con i bordi invisibili) e poi premete il tasto di destra appariranno i tre menu dello Workbench; clickando nell'output o nella list window e attivando nuovamente i menu vedrete apparire quelli del Basic, in numero di quattro.

Torniamo al nostro Basic. Potete chiudere, se lo desiderate, una qualunque delle due finestre; potrete poi riaprirla con l'opzione Show Output o Show List del menu Windows. Queste direttive riaprono la finestra in questione se non esiste o la spingono davanti alle altre se è già aperta. Chiudendo entrambe le finestre si esce dal Basic. Capita spesso di voler riaprire o riportare in superficie la list window, quindi è stato creato un menu short*cut* (scorciatoia) che consiste nel premere i tasti Amiga sinistra - L che sono equivalenti alla selezione di Show List da menu. Altri utili e spesso usati shortcut sono Amiga sinistra - R e Amiga sinistra - punto, che significano rispettivamente run (fai partire il programma) e break (fermalo) e corrispondono alle opzioni Start e Stop del menu Run. Il break può essere ottenuto anche con il più standard CTRL - C. A proposito, non abbiate mai timore di danneggiare l'Amiga con operazioni errate; il peggio che vi possa capitare è costituito dal blocco temporaneo del computer che costringe a spegnerlo per mezzo minuto, o dalla cancellazione dei dischi (ma voi usate sempre e solo le copie, non è vero?). L'unico modo per rovinare un computer attraverso la sua tastiera consiste nello sfondarla a martellate. Il megacomputer che va irrimediabilmente in fumo perché gli si è chiesto di calcolare il numero più grande del mondo è un mito esistente solo nei film poco accurati e nella fantasia popolare.

Selezionate l'output window e digitate "print 123" (senza le virgolette e seguito, naturalmente, dal tasto Enter o Return che dir si voglia); nella stessa finestra compare la scritta 123 seguita da Ok. Il computer ha eseguito all'istante l'ordine di stampare il numero 123 (to print significa stampare, ma non illudetevi che il computer capisca il normale inglese; la comprensione del linguaggio naturale è ancora uno dei temi di punta delle ricerche sull'intelligenza artificiale). Digitate "print 246" nella finestra di list (con tanto di Enter, che d'ora in avanti sarà sottinteso) e potrete vedere che non accade assolutamente nulla. Premete Amiga - R (cioè tenete abbassato il tasto Amiga di sinistra, date un colpetto alla R e rilasciate il primo tasto); la scritta 246 apparirà nell'output window. Ogni volta che date il run con Amiga - R appare un altro 246. L'Amiga ha quindi memorizzato l'ordine e lo può rieseguire quante volte volete.

Ogni ordine digitato nell'output window viene eseguito subito e dimenticato, mentre quelli inseriti nella finestra di list entrano a far parte del programma *corrente* che resta in memoria fino a nuovo ordine (o finchè qualcuno non inciampa nel cavo dell'alimentazione, staccando la spina dalla presa di corrente). Avete appena scritto il vostro primo programma in Basic e avete anche imparato come farlo eseguire; se volete rivederne il listato (cioè l'insieme delle istruzioni che lo costituiscono, in questo caso una sola) premete Amiga - L (la finestra di list ha sempre il buon gusto di togliere l'incomodo al momento del run e bisogna quindi farla riapparire). Vogliamo registrare su disco questo programmino in modo che non svanisca spegnendo l'Amiga? Entrate nel menu Project e attivate il comando Save; apparirà un requester che vi chiede il nome con cui battezzare il nuovo nato. Clickate all'interno del rettangolo allungato per attivarlo e digitate un nome di vostro gusto, poi premete (come sempre) il tasto di immissione (l'Enter, e che nessuno si sogni di chiamarlo Immis come qualcuno orrendamente fa, pena la condanna alla traduzione di tutti i termini tecnici computereschi in latino; tale lingua non è poi morta in misura maggiore del vocabolo Immis, che si aggira come uno zombie nel limbo dei neologismi defunti prima di nascere).

A questo punto il programma è stato salvato su disco. Attiviamo New dal menu Project e lo vedremo sparire dalla list window, sintomo inequivocabile della sua cancellazione dalla memoria RAM del computer; scegliamo Open dallo stesso menu, digitiamo il nome del programma nell'ormai consueto rettangolo, premiamo l'Enter ed il programma ricomparirà dopo essere stato ricaricato da disco. Per questa volta abbiamo finito; questa è stata più che altro una lezione introduttiva per acquisire una certa familiarità con il sistema nel suo complesso. Dalla prossima volta entreremo nel vivo dei concetti della programmazione.

La grafica

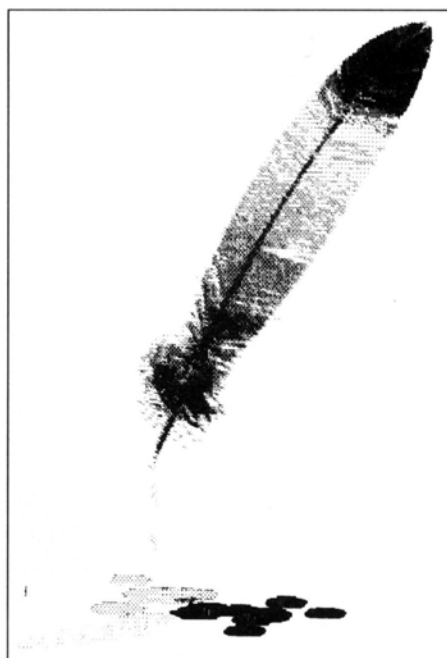
Riprendiamo il discorso riservato a chi

sa già cosa significa programmare in Basic. La grafica è un argomento di importanza fondamentale per ogni computer e per l'Amiga in particolare, che ne fa il proprio punto di forza. Purtroppo la potenza e la flessibilità dell'Amiga in campo grafico non sono pienamente sfruttabili in Basic, come vedremo meglio più avanti. Il numero di modi grafici che il nostro beneamato computer possiede è rilevante; quelli raggiungibili da Basic sono 18 su 38.

Per fare della grafica occorre innanzitutto uno schermo con una o più finestre al suo interno; si possono anche creare più schermi (fino a quattro in Basic) ognuno dei quali può essere spostato con il mouse davanti o dietro agli altri, come si fa con le finestre. L'unico schermo sempre presente è quello dello Workbench, al cui interno è inizialmente aperta l'usuale output window che tutti i programmi Basic privi di particolari requisiti grafici utilizzano per l'input-output. Esso ha una risoluzione di 640x256 pixel con quattro colori; se tale configurazione non vi soddisfa dovrete esplicitamente aprire un nuovo schermo con le caratteristiche desiderate, altrimenti potete usare quello già esistente. Lo statement SCREEN n,x,y,npl,m apre uno schermo di dimensioni x per y, costituito da npl piani di bit (e quindi capace di 2^npl colori scelti tra 4096), nel modo grafico m e al quale viene assegnato il numero di identificazione n, che deve essere compreso tra uno e quattro. Il parametro m, anch'esso compreso tra uno e quattro, determina il modo grafico in base alla seguente codifica: se m è maggiore di due lo schermo è interlacciato (alta risoluzione verticale), altrimenti no; un valore pari indica l'alta risoluzione orizzontale, uno dispari la bassa. In realtà il modo grafico è deducibile dalle dimensioni dello schermo, in base alla seguente formula semibooleana, dove il vero vale -1 ed il falso 0:

```
m=1-(x>450)-2*(y>300)
```

Ad esempio, supponiamo di desiderare uno schermo 640x200:

```
m=1-(640>450)-2*(200>300)
m=1-(vero)-2*(falso)
m=1-(-1)-2*(0)
m=1+1-0
m=2
```

Il parametro npl può spaziare da uno a cinque; il valore più basso corrisponde a due soli colori disponibili, il più alto a 32 (ad ogni aumento di un'unità da parte di npl il numero di colori raddoppia).

Può essere istruttivo, nonchè utile in pratica, sapere quanta RAM consuma ogni schermo; l'occupazione di memoria in byte è data dalla seguente formuletta:

```
mem=INT((x+15)/16)*2*y*npl
```

Lo schermo dello Workbench, ad esempio, che possiede una grafica 640x256 con due piani di bit, richiede 40K. Il più potente modo grafico ottenibile, con una risoluzione pari a 640x512 pixel e 16 colori (quattro bitplane), consumerebbe 160K. L'uso del condizionale è reso opportuno dalla scarsa probabilità di riuscire a reperire un simile quantitativo di RAM libera in un sistema inespanso quando il Basic è in memoria. Uno schermo creato dall'utente può essere eliminato con lo statement SCREEN CLOSE n.

Per utilizzare uno schermo è necessario aprire al suo interno almeno una finestra. Il comando che lo consente è WINDOW n, title$, (x1,y1)-(x2,y2), type, screen; n è il numero che viene assegnato alla finestra così creata; il numero 1 è riservato alla finestra di output iniziale, quella che appare da sola dopo aver caricato il Basic. Title$ è una stringa opzionale che appare nella barra superiore della finestra. Le coppie di coordinate, anch'esse opzionali, localizzano rispettivamente l'angolo in alto a sinistra e quello in basso a destra della window rispetto all'origine dello schermo; se si decide di non specificarle i loro valori di default prevedono che la finestra occupi l'intero schermo. Type è un parametro alquanto complesso, essendo la somma di cinque coefficienti. Per calcolare il valore che più si adatta alle vostre esigenze partite da zero e sommate 1 se volete dotare la finestra del sizing gadget, sommate ancora 2 se la finestra potrà essere spostata con il mouse agendo sulla sua barra, 4 se desiderate front e back gadget, 8 per il close gadget ed infine 16 se volete che il contenuto della finestra non vada perduto quando viene coperta temporaneamente da un'altra finestra. Quest'ultima caratteristica, se attivata, consuma RAM in misura proporzionale alle dimensioni della finestra; se così non fosse tutti la richiederebbero in ogni caso e non avrebbe senso lasciare tale scelta al programmatore. Se poi avete deciso di sommare sia l'1 che il 16 sappiate che ciò vi costerà tanta RAM quanta ne consuma lo schermo stesso, come se la finestra avesse sempre le massime dimensioni.

E' opportuno illustrare i concetti tutt'altro che coincidenti di active window e di *current output window*, sui quali il manuale non si sofferma a sufficienza; direi anzi che non vi si ferma proprio. La finestra attiva è quella selezionata con il mouse, l'unica in un dato istante ad avere la barra del titolo nitida e capace di ricevere un qualunque input dall'utente. La finestra di output è quella alla quale sono diretti tutti gli output, e non è quindi selezionabile dall'utente come avviene per la finestra attiva, bensì dal programma, tramite l'istruzione WINDOW OUTPUT n. Esiste poi la direttiva WINDOW n, non ben documentata dalla reference guide (al punto che l'autore del presente articolo, pur avendo l'abitudine di leggere a fondo i manuali, l'ha scoperta solo per caso), il cui scopo consiste nel portare in superficie la finestra specificata. Spesso si ha l'esigenza di rendere attiva una finestra da programma; purtropo non esiste comando Basic capace di assolvere questo compito e bisogna ricorrere a chiamate dirette alla ROM. Ma di questo parleremo un'altra volta.

MERCATINO

Vendo **C 64,** registratore C2N, driver 1541, stampante 802, circa 2000 programmi e manuali. Inoltre walkman stereo Hitachi. Il tutto, in ottime condizioni, a L. 1.000.000. Claudio Borgonovi - Via Antonio Coppi, 6 - 00179 Roma - Tel. (06) 7828276

Compro **monitor 1081 per Amiga**, preferibilmente modello vecchio. Cerco possessori di Amiga in zona Carpi per scambio di programmi. Enrico Filippi - Via F.lli Saguatti, 4 - 41010 Carpi (MO) - Tel. (059) 660253

Cambio **game e utility per C 64**, solo su disco. Inviare lista a: Giuseppe Di Lello - Corso Europa, 13 - 66054 Vasto (CH) - Tel (0875) 60393

Vendo **Amiga 2000** più doppio drive, nuovo ancora in garanzia, a sole L. 2.000.000. Telefonare dopo le ore 19,30 a Piero Bertoldo - Via Lombardia, 1 - 25098 Verolanuova (BS) - Tel. (030) 931874

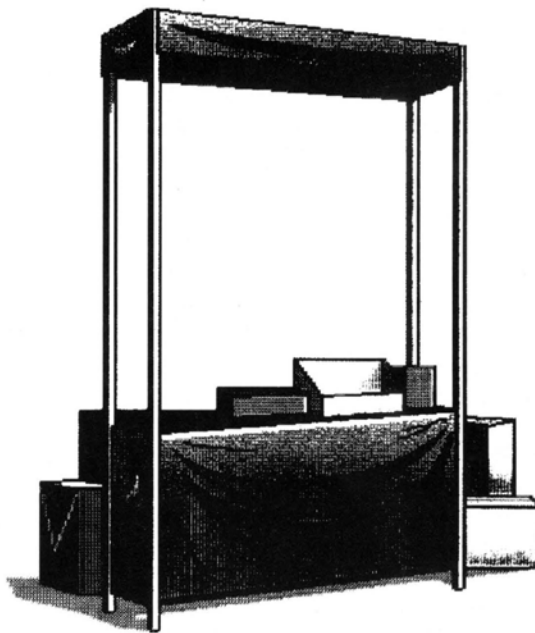
Vendo **C 64 New** più registratore, joystick, guida riferimento, light pen, oltre 200 giochi. Tutto in perfette condizioni a L. 390.000. Rivolgersi a Diego Lorenzini - Via XX Settembre, 33 - 13011 Borgosesia (VC) - Tel. (0163) 22909

Possessori di **Vic 20**, volete cambiare programmi? Scrivete unendo la vostra lista a : Mauro Griffo - Casella postale 17024 - 17024 Finale Ligure (SV)

Vendo **Amiga 1000,** sistema completo con imballaggi originali, 300 programmi e molti manuali, al migliore offerente. Telefonare ore pasti a Alessandro Serafini - Via Salvolini, 3 - 60129 Ancona - Tel. (071) 31273

Vendo **C 64** praticamente nuovo con un manuale guida originale e 30 tra i migliori videogame a L. 290.000 non trattabili. Telefonare dopo le 18 a Fabrizio Rocchi - Via Carducci , 19 - 20092 Cinisello Balsamo (MI) Tel. (02) 6184760

Vendo **C 64**, registratore C2N, due joystick, 700 programmi (utility, didattici, creativi, giochi), quattro cartucce, duplicatore, allineatore, turbo e manuali e riviste a L. 450.000. Contattare Roberto Azzué - Via S. Giacomo, 104 - 80013 Casalnuovo (NA) - Tel. (081) 8422462

Cambio per **Amiga** qualunque tipo di programma. Per informazioni scrivere o telefonare a: Santi Mondo - Via Orsa Maggiore, 53 - 98057 Milazzo(ME) Tel. (090) 9284863

Compro stampante Commodore Mps 1000. Cambio e/o acquisto programmi per Amiga. Rivolgersi a: Gianni Cottogni - Via Strambino, 23 - 10010 Carrone (TO) - Tel. (0125) 712311.

Vendo **Commodore 128D con drive 1571** incorporato, monitor monocromatico, stampante Pd 80, joystik da 8, registratore 1516, programmi e manuali. Garanzia dicembre '88. Il tutto ancora imballato a L. 1.800.000. Telefonare ore pasti a: Massimiliano Aloisi - Via Modena, 21 - Cutrofiano (LE) - Tel . (0836) 662443.

Vendo **Amiga 1000** tastiera americana a L. 1.200.000, monitor stereo a L. 400.000, drive esterno a L. 200.000. Prezzi trattabili. Telefonare ore pranzo a: Valerio Bonetti - Via Torritano, 13 - 31100 Treviso - Tel. (O422) 22350.

Cambio **programmi** solo su disco con utenti C 128 o C 64. Cerco e offro massima serietà. Rivolgersi a: Enrico Gaspari - Via Arezzo, 35 - 63040 Folignano (AP) - Tel. (0736) 492026

Cambio **programmi per Amiga**. Inviare le proprie liste a: Roberto De Chaud - Via Sant'Elena, 198 - 16153 Genova-Sestri Ponente - (Tel. 010) 603726.

Cambio **software per Amiga**. Non vendo e non compro. Massima serietà e celerità. Lista di circa 600 programmi. Mario Muré - Via Stella, 1 - 4100 Modena - Tel. (059) 214841

Vendo **modem** Spider 1200 con autodial e autoanswer, linguaggio At, uscita Rs-232 normalizzata; standard V.21, V.22, Bell 103, Bell 212. Roberto Moretti - Strada Torino, 73 - 10024 Moncalieri (TO)

Inviare questo coupon o una sua fotocopia a:
"Mercatino" AMIGA MAGAZINE
Gruppo Editoriale Jackson
via Rosellini, 12 - 20124 MILANO

Cognome ______ Nome ______

via ______ n. ____ C.A.P. ______

Città ______ tel. ______

Firma ______ Data ______

AM 5

Andrea Bigiarini - Pierluigi Cecioni - Marco Ottolini
Il Manuale di AMIGA
HOME E PERSONAL COMPUTER
Rita Bonelli - Massimiliano Lunelli
AMIGA 500
Guida per l'utente
HOME E PERSONAL COMPUTER
GRUPPO EDITORIALE JACKSON
David Lawrence - Mark England
AMIGA Handbook
GRUPPO EDITORIALE JACKSON
GRUPPO EDITORIALE JACKSON

di Paolo Russo

Questa puntata è un po' sui generis per più d'un motivo. La trattazione delle istruzioni del microprocessore cede temporaneamente il passo alle direttive di assembly, che non hanno importanza in fase di esecuzione bensì a monte, durante l'assemblaggio. Alla fine della lezione viene presentato Print&Input, un ausilio alla programmazione che, consentendo in modo assai semplice di realizzare un po' di input-output, risulta utile soprattutto a chi sta muovendo i primi passi in questo affascinante mondo dell'Assembly, l'unico dove, a seconda del punto di vista, veramente tutto (o quasi nulla) è al tempo stesso possibile.

Cos'è una direttiva di assembly

In ogni sorgente (con questo nome si identifica, come spiegato in precedenza, il file contenente il listato, in contrapposizione al codice oggetto, contenente il programma assemblato), nel campo delle istruzioni (opcode field), possono trovarsi comandi diretti al microprocessore o all'assembler. L'istruzione MOVE, ad esempio, viene eseguita dal 68000 mentre la direttiva SET vista nella puntata precedente è rivolta all'assembler e non genera codice macchina. L'assembler traduce le istruzioni del 68000 comportandosi come un compilatore, ma esegue direttamente come un interprete quei comandi, che non fanno parte dell'Assembly MC68000, e che influenzano piuttosto il modo in cui le successive istruzioni dovranno essere tradotte. E' d'uopo un esempio.

La linea: Tre SET 3 assegna il valore 3 alla label di nome Tre. D'ora in avanti quando l'assembler incontrerà la parola Tre saprà che in realtà essa significa 3 e sostituirà tale valore al nome della label. L'istruzione ADDQ.W #Tre,D2 sommerà quindi 3 al registro D2. La parola SET non fa parte del set di istruzioni del 68000 ed è rivolta all'assemblatore, il quale la esegue all'istante inserendo la parola Tre con il valore 3 nella sua tabella delle label (ogni assembler possiede una tabella dove memorizza le label a mano a mano che le incontra nel listato).

La direttiva più usata è EQU; anch'essa, come SET, assegna un valore ad una label, ma, a differenza di SET, l'assegnazione è permanente. Una label che viene definita da un'EQU non può in futuro ricevere alcun altro valore; ogni tentativo in tal senso verrà interpretato dall'assembler come un errore. Solitamente, potendo scegliere, si usa EQU, che garantisce l'impossibilità di utilizzare per sbaglio lo stesso nome per due label differenti. Un altro pregio (storico) di questa direttiva consiste nell'essere supportata da ogni tipo di assembler per ogni processore, perfino per quelli immaginari (come accade nelle Core Wars) mentre talvolta SET risulta non implementata.

CORSO DI "ASSEMBLY"

Un'avvertenza, prima di continuare; le direttive di assembly possono variare, da un punto di vista sia sintattico che funzionale, quando si cambia assembler (eccezion fatta per EQU). In questo corso si è deciso di adottare l'assembler standard dell'Amiga prodotto dalla Metacomco, che risulta affidabile anche se paurosamente lento ed affamato di RAM. "Standard" significa che è espressamente citato e documentato dai manuali dell'Amiga e che di conseguenza molti altri assembler cercano di essere compatibili con esso.

Byte, word, long word di dati

Qualunque programma non banale contiene quasi sempre una sezione contenente variabili e dati già inizializzati, ad esempio tabelle di costanti. La direttiva DC (Define Constant) consente di inserirli nel listato (un po' come la parola DATA in Basic, ma non è proprio la stessa cosa, perché DC viene usata anche per definire le variabili). La parola DC è sempre seguita da .B, .W o .L ed accetta un qualunque numero di parametri (esempio: DC.W 3,5,257,-14). Vediamo un esempio, dove una stringa di 14 byte (vedi StrLength) incorporata nel programma viene copiata in un'altra zona di RAM, il cui indirizzo è contenuto nella variabile Address:

```
Copia    MOVE.W StrLength(PC),D0
         LEA String(PC),A0
         MOVE.L Address(PC),A1
         BRA.S EndLoop
Loop     MOVE.B (A0)+,(A1)+
EndLoop DBRA D0,Loop
         ...
Address DC.L 120000
StrLength DC.W 14
String   DC.B 'Salve a tutti!'
```

Per inciso, l'istruzione LEA carica un indirizzo in un registro indirizzi. Notate a tal proposito la differenza tra la seconda e la terza linea dello spezzone appena visto: la linea LEA String(PC),A0 pone *l'indirizzo* della stringa in A0 mentre MOVE.L Address(PC),A1 pone in A1 il *contenuto* della long word all'indirizzo Address. Il "(PC)" segnala che gli indirizzi specificati sono da intendersi relativi (16 bit) al program counter; se togliessimo il "(PC)" dalle linee precedenti gli indirizzi diverrebbero assoluti (32 bit) ed il programma si allungherebbe un pochino. La presenza del "(PC)" non è quindi critica nel nostro caso, ma talvolta lo è (spesso è proibita, ogni tanto è obbligatoria, a seconda delle situazioni. In futuro l'argomento verrà trattato molto più in dettaglio).

La direttiva DS (Define Storage) serve a riservare uno spazio per volumi di dati relativamente grossi (e non inizializzati). Ad esempio DS.W 100 crea una zona libera di 100 word (200 byte) nel punto in cui è invocata (è pressappoco equivalente a 100 istruzioni DC.W 0 consecutive).

Le label locali

Le label locali, ad essere precisi, non sono direttive ma, per l'appunto, label. Ad ogni modo esse non fanno parte delle convenzioni dell'Assembly MC68000 e sono invece supportate dall'assembler standard, quindi si è ritenuto più logico parlarne a proposito delle direttive di assembly. Una label è locale se il suo nome è un numero seguito dal segno del dollaro; ad esempio SALVE è normale mentre 13$ è locale. Le label di quest'ultimo tipo esistono soltanto tra due label normali, nel senso che la definizione di una label normale costringe l'assembler a dimenticare tutte quelle locali definite fino a quel momento. Essendo la questione in realtà assai semplice, ma complessa da esprimere a parole, vediamone un esempio:

```
Fill0  MOVEQ #99,D0
1$ CLR.B (A0)+
         DBRA D0,1$
         RTS
Fill1  MOVEQ #99,D0
1$ ST (A0)+
         DBRA D0,1$
         RTS
```

Questo breve spezzone di programma contiene due subroutine, di nome Fill0 e Fill1; la prima riempie di zeri una zona di memoria puntata da A0 e lunga 100 byte, mentre Fill1 fa esattamente l'opposto, assegnando il valore -1 a quegli stessi byte (l'istruzione ST è l'opposto di CLR.B e pone 255, ossia -1, nel byte specificato come parametro). La label 1$ viene definita una prima volta nella seconda linea, allo scopo di marcare l'inizio di un loop; alla fine di tale loop (che è lungo solo due istruzioni) avviene un salto condizionato (DBRA

D0,1$) all'inizio dello stesso. La label 1$ viene poi ridefinita e riutilizzata una seconda volta, a distanza di poche linee, nella seconda subroutine, ma ciò è lecito e non confonde l'assembler in quanto la label Fill1 viene definita tra il primo e il secondo utilizzo di 1$, fungendo così da confine. Si può dire che la prima label 1$ ha valore solo nelle linee comprese tra Fill0 e Fill1 mentre la seconda label 1$ è valida da Fill1 fino alla prossima label normale.

Considerando la situazione dal punto di vista dell'assembler, ogni volta che nel programma compare un'istruzione (come DBRA) che fa riferimento ad una label definita da qualche altra parte, per tradurre il tutto in linguaggio macchina è necessario cercare il punto esatto marcato da quella label, ma tale ricerca, nel caso delle label locali, non si spinge mai oltre il confine segnato da una qualunque label normale, sia quando si cerca in avanti che all'indietro. Di conseguenza, nell'esempio precedente, quando l'assembler deve tradurre in codice macchina la prima istruzione DBRA D0,1$, si rende conto che per farlo deve prima trovare la definizione della label 1$; dal momento che il DBRA si trova in una sezione di programma compresa tra le label Fill0 e Fill1, la ricerca di 1$ avviene nello stesso ambito, ed essendo tale label definita una sola volta in quell'intervallo non esiste possibilità d'errore. La stessa cosa avviene quando l'assembler ha a che fare con la seconda istruzione DBRA, con l'unica differenza che l'intervallo di ricerca si estende da Fill1 in poi.

Vediamo adesso qualche esempio sbagliato, dove l'assembler si fermerebbe segnalando un errore, assieme alla relativa versione corretta:

```
* SBAGLIATO
Somma MOVEQ #0,D1
       BRA.S 1$
Loop   ADD.W (A0)+,D1
1$     DBRA D0,Loop
       RTS

* GIUSTO
Somma MOVEQ #0,D1
       BRA.S 1$
2$     ADD.W (A0)+,D1
1$     DBRA D0,2$
       RTS

* SBAGLIATO
       MOVEQ #99,D1
1$     CLR.W (A0)+
       DBRA D1,1$
       MOVEQ #99,D1
1$     MOVE.W #1,(A0)+
       DBRA D1,1$

* GIUSTO
       MOVEQ #99,D1
1$     CLR.W (A0)+
       DBRA D1,1$
       MOVEQ #99,D1
2$     MOVE.W #1,(A0)+
       DBRA D1,2$
```

Il primo errore consisteva nella presenza di una label normale tra la definizione di una locale ed il suo utilizzo; l'assembler avrebbe segnalato la presenza di un "undefined symbol in operand field" nella linea BRA.S 1$. Il secondo errore era l'opposto del precedente: la label 1$ era stata definita due volte senza l'interposizione di alcuna label normale.

Allineamento a indirizzo pari

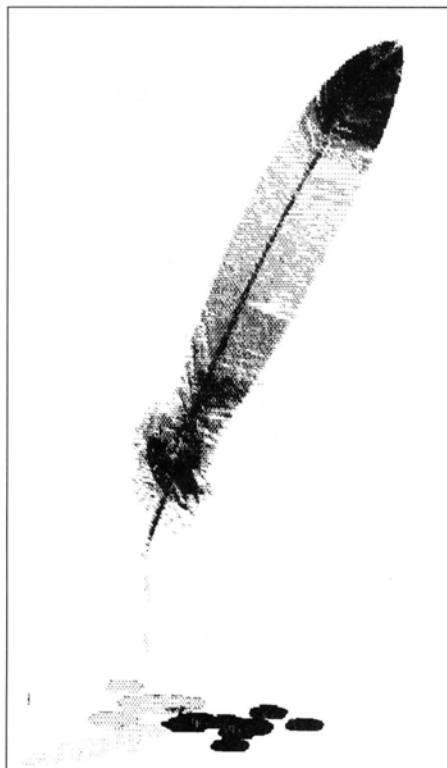
Tutte le istruzioni del 68000 devono trovarsi ad un indirizzo di memoria pari. La prima istruzione di un programma lo è sempre, e tutte le istruzioni del 68000 hanno lunghezza pari, così che in teoria non dovrebbe mai accadere che un'istruzione possa finire ad un indirizzo dispari. Tuttavia, assai spesso i dati e le variabili di un programma sono interposti tra le sezioni eseguibili e può accadere che alcuni dati (tipicamente stringhe) abbiano lunghezza dispari. In questi casi bisognerebbe calcolare la lunghezza della stringa e, se questa fosse dispari, sarebbe necessario aggiungergli un byte "cuscinetto" in coda. Tutto ciò non è in realtà necessario grazie alla presenza dell'istruzione CNOP 0,2 (Conditional Nop = No Operation) che, posta subito dopo una stringa, aggiunge un ulteriore byte solo se necessario.

Con altri parametri tale istruzione svolge funzioni differenti; ad esempio CNOP 0,4 allinea ad un indirizzo multiplo di quattro.

Assemblaggio condizionato

Capita talvolta che si desideri assemblare una certa parte di un programma solo se risulta verificata una particolare condizione, ad esempio se una certa label non è ancora stata definita o se supera un certo valore. Quasi certamente il 95% dei lettori starà pensando che *mai* nella loro vita potrebbe accadere loro di aver bisogno di una cosa del genere. Ebbene, prima della fine di questa lezione la maggior parte di essi avrà cambiato idea.

La direttiva IFcc consente di subordinare l'assemblaggio di un tratto di programma al verificarsi di una condizione, che può essere di tre tipi:
1) Esistenza di una label: IFD label (If Defined), IFND label (If Not Defined).
2) Valore di un'espressione di costanti: IFEQ, IFNE, IFGT, IFGE, IFLT, e IFLE, tutti seguiti dall'espressione in questione. Esempio: se vogliamo controllare la condizione che la label zap sia uguale a 3 la direttiva sarà IFEQ zap-3.
3) Coincidenza di due stringhe: IFC 'string1','string2' (If Coincident), IFNC 'string1','string2' (If Not Coincident). Questa direttiva è utile per eseguire dei test sui parametri di una macro (cosa sia quest'ultima è spiegato più avanti).

In ogni caso, la direttiva ENDC (End Condition) deve essere posta al termine del blocco di linee da assemblarsi condizionatamente, in modo simile agli statement IF ed END IF dell'AmigaBasic. L'EL-

SE, purtroppo, non è implementato. Il THEN non esiste perché è sempre sottinteso. Le coppie IFcc - ENDC possono essere nidificate, ossia è possibile inserire al loro interno altre coppie simili; l'assembler terrà il conto del numero di IFcc in sospeso.

Le macro

Dopo aver programmato in Assembly MC68000 per un certo tempo ci si accorge che è molto frequente dover salvare tutti i registri sullo stack e doverli successivamente ripristinare. L'istruzione che realizza un push su così vasta scala, come risulta dalle precedenti lezioni, è MOVEM.L D0-D7/A0-A6,-(A7), mentre il pop corrispondente è MOVEM.L (A7)+,D0-D7/A0-A6. Entrambe le istruzioni sono alquanto scomode da digitare. Sarebbe bello che esistessero istruzioni dal nome più corto, tipo PUSHREGS e POPREGS, con la medesima funzione. Grazie alle macro ciò è possibile. Inseriamo le seguenti linee all'inizio di ogni programma e il nostro desiderio si tramuterà in realtà:

```
PUSHREGS MACRO
         MOVEM.L D0-D7/A0-A6,-(A7)
         ENDM
POPREGS  MACRO
         MOVEM.L (A7)+,D0-D7/A0-A6
         ENDM
```

D'ora in avanti potremo usare le pseudoistruzioni PUSHREGS e POPREGS come sinonimo delle scomode notazioni viste in precedenza; ogni volta che l'assembler incontrerà, nel resto del listato, la parola PUSHREGS la sostituirà automaticamente con l'istruzione MOVEM.L corrispondente, prima di tradurre il tutto in codice macchina. La sostituzione opera a livello di stringhe di testo sorgente. Se poi provassimo a disassemblare il codice macchina così ottenuto non troveremmo traccia dell'inesistente istruzione PUSHREGS; al suo posto troveremmo il MOVEM.L. Non si tratta quindi neppure di subroutine; abbiamo appena creato due macro.

S'intende che le macro hanno un campo di applicazione ben più vasto di quello appena visto, ed in questa puntata ne vedremo un esempio di utilizzo alquanto esteso. Una caratteristica molto importante delle macro è la possibilità del passaggio di parametri, rappresentati all'interno della macro stessa come una barra inclinata a sinistra (backslash) seguita da un numero da uno a nove. I parametri possono essere qualunque cosa venga digitato dopo il nome della macro. Vediamo un esempio:

```
INC  MACRO
     ADDQ.W #1,\1
     ENDM
     ...
     INC D3
     INC A0
     INC (A2)+
     INC 16(A5)
```

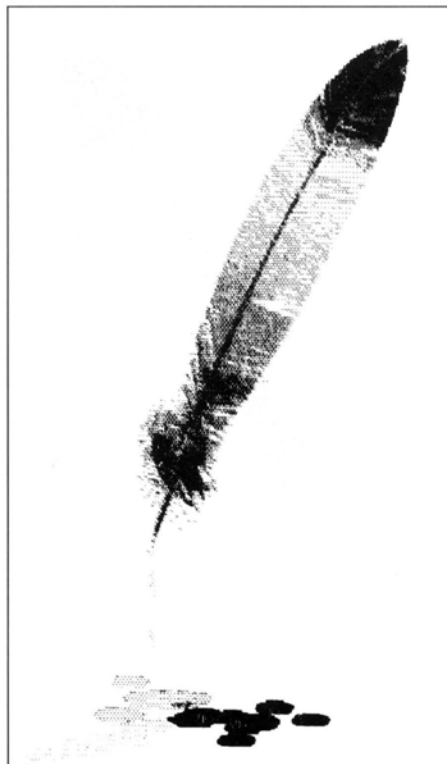
Nel corpo della macro il simbolo \1, parametro formale, viene di volta in volta sostituito da D3, A0, (A2)+ e 16(A5), parametri effettivi. Ad ogni macro possono essere passati fino a nove parametri, da \1 a \9; il primo si chiama sempre \1, il secondo \2 e così via. Nella chiamata della macro essi devono essere separati da virgole. Un parametro non passato equivale ad una stringa nulla. Se Zip è una macro e noi digitiamo Zip #2,D3,,,A0 i soli parametri non nulli sono il primo, il secondo e il quinto. Un simbolo speciale, NARG (Number of Arguments), usato all'interno di una macro restituisce il numero di parametri passati alla stessa (nel caso di Zip varrebbe cinque).

Tra le direttive MACRO ed ENDM può essere presente un numero arbitrario di linee di programma; è possibile, naturalmente, richiamare una macro dall'interno di un'altra macro, anche se esiste un limite al numero di annidamenti di questo genere (di solito si aggira intorno a dieci). Non è però possibile definire una macro all'interno di un'altra. Esiste anche la direttiva MEXIT, di uso un po' raro, che provoca l'uscita immediata dalla macro (ricorda vagamente l'EXIT SUB dell'AmigaBasic, ma ricordiamoci che le macro non sono sottoprogrammi) e che di conseguenza ha senso solo se preceduta da un IFcc (un MEXIT incondizionato equivarrebbe a un ENDM). Le macro sono molto usate nei file di testo che si prevede siano inclusi in altri (vedi prossimo paragrafo).

La direttiva INCLUDE

Provate ad esaminare le prime righe di un qualunque programma scritto in C per l'Amiga; quasi certamente la sezione iniziale trabocca di comandi #include. La situazione non è molto diversa in Assembly, dove è reputato normale (per gli amighevoli standard) inserire una interminabile sfilza di INCLUDE all'inizio del listato. Questa importantissima direttiva consente la fusione temporanea (per il tempo necessario all'assemblaggio) di più file sorgenti. Supponiamo ad esempio di trovare tanto utili le macro PUSHREGS e POPREGS, che abbiamo definito poco fa, da volerle presenti in ogni nostro programma. Sarebbe davvero scomodo doverne digitare la definizione ogni volta; tale seccatura annullerebbe il vantaggio ottenuto. Possiamo allora creare un file, di nome Stackregs, contenente unicamente le definizioni delle suddette macro, e ogni volta che vorremo impiegarle in un nostro programma sarà sufficiente inserire la linea INCLUDE "Stackregs" all'inizio dello stesso. S'intende che, durante la fase di assemblaggio, il file Stackregs deve trovarsi nella directory corrente; in caso contrario esiste sempre la possibilità di specificare un pathname completo tra le virgolette del comando INCLUDE.

E' interessante notare che quasi ogni linguaggio esistente prevede la possibili-

tà dell'inclusione di file esterni. Il testè citato #include del C è solo un'esempio; anche il TurboPascal prevede qualcosa di simile, per non parlare dell'ultima generazione di linguaggi strutturati che fa dell'inclusione una componente fondamentale della sua filosofia. Anche l'AmigaBasic possiede i comandi MERGE e CHAIN MERGE, che rappresentano per un interprete l'equivalente approssimativo dell'inclusione tipica dei compilatori.

Print&Input, finalmente!

La precedente puntata di questo corso si era chiusa con la constatazione delle difficoltà inerenti alla realizzazione di un print o di un input in Assembly, e soprattutto con la promessa di una soluzione al suddetto problema. Promessa mantenuta: ecco a voi Print&Input, il cui listato relativamente lungo, oltre che pubblicato in queste pagine, è ovviamente presente nel disco che da sempre integra la parte cartacea di Amiga magazine, pronto per essere usato nei vostri programmi con un INCLUDE.

Print&Input contiene parecchie subroutine e macro, la maggior parte delle quali viene usata internamente; le macro PRINTF, PRINT e INPUT sono invece accessibili a chiunque. Iniziamo dalla prima, che è anche la più complessa. I parametri che la seguono (in numero massimo di nove) sono associati a due a due; il primo elemento di ogni coppia descrive il tipo del secondo elemento, che rappresenta il vero e proprio "oggetto" da stampare. I tipi possibili sono quattro: S=stringa, B=byte, W=word, L=long word. Esiste poi l'opzionale parametro finale N, l'unico a non far coppia con nessuno, che significa "vai a capo" (tecnicamente, si tratta di una specie di codice di controllo che stampa un linefeed, simile al \n del C). Le stringhe devono essere comprese tra i simboli < e > e, purtroppo, non possono contenere l'apostrofo per ragioni tecniche pressochè insormontabili. Vediamo subito qualche esempio:

```
PRINTF S,<Salve, mondo!>,N
PRINTF L,D0,N
PRINTF S,<Il registro D0 vale >,L,D0,N
PRINTF W,D1,W,D2,B,D5,N
```

Non sussiste alcun dubbio sulla scomodità del dover sempre specificare il tipo del parametro. E' stata quindi creata una versione semplificata e, in un certo qual modo, intelligente della macro PRINTF, di nome PRINT; questa macro non richiede l'uso dei codici B, W, L, S o N, è in grado di distinguere automaticamente se un parametro rappresenta il nome di un registro o no e si regola di conseguenza, stampando come long word i registri e come stringhe qualunque altra cosa. Alla fine della stampa è sempre sottinteso il ritorno a capo. Vediamo quindi che aspetto assumono gli stessi quattro esempi visti sopra, utilizzando però PRINT al posto di PRINTF:

```
PRINT <Salve, mondo!>
PRINT D0
PRINT <Il registro D0 vale >,D0
PRINT D1,D2,D5
```

In realtà l'ultimo esempio non è uguale alla precedente versione, in quanto i registri D1, D2 e D5 vengono adesso stampati come long word, il che può non essere affatto ciò che vogliamo. I segni < e > in realtà non sono tipici delle stringhe, ma devono essere usati quando si passa ad una macro un parametro che contiene uno spazio. Anche in ambiente CLI avviene qualcosa di simile: i nomi di file contenenti spazi devono essere racchiusi tra virgolette.

Last but not least, la macro INPUT realizza esclusivamente input numerici formato long word. Esempio:

```
INPUT <Dammi la X >,D1,<Dammi la Y >,D2
```

Qualche considerazione

Date una rapida occhiatina al listato di Print&Input; vedrete che pullula di macro e assemblaggi condizionati, per non parlare del fatto che per utilizzarlo in un programma è necessario un INCLUDE "Print&Input".

Coloro che dubitavano dell'utilità di simili direttive si saranno ricreduti. I più curiosi sui meccanismi interni gradiranno sapere che le macro PRINTF, PRINT e INPUT generano ognuna poche istruzioni, tra le quali una chiamata alle subroutine che eseguono materialmente l'input-output.

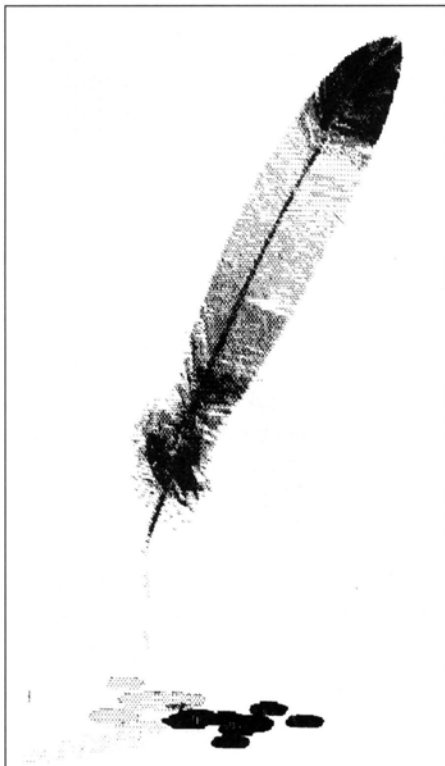
L'occupazione di memoria da parte delle chiamate di macro è quindi limitata al minimo indispensabile.

Non tentate, ad ogni modo, di comprendere il funzionamento interno di Print&Input, che implica una discreta conoscenza del sistema operativo dell'Amiga (Exec e Dos, per la precisione).

Potete piuttosto rivolgere una certa attenzione ai listati di Eratostene e Pitagora, due programmini che fanno uso di Print&Input; il primo implementa (in modo volutamente non ottimizzato) il ben noto crivello di Eratostene (il famoso benchmark Sieve) per trovare i numeri primi; il secondo calcola l'ipotenusa di un triangolo rettangolo dati i cateti, con la limitazione che tutti i numeri devono essere interi, e sulla spaventosa utilità di questo programma è superfluo spendere ulteriori parole.

Speriamo almeno che raggiungano entrambi il fine didattico che ne è alla base.

Per quanto riguarda le direttive di Assembly, ce ne sarebbero ancora diverse decine da trattare, ma a mio personale ed insindacabile giudizio la loro utilità è virtualmente inesistente, eccezion fatta per la parola opzionale END che marca la fine di un programma e per le direttive XDEF e XREF che interagiscono pesantemente con il linker più che con l'assembler e meritano un discorso a parte.

TRA STRUTTURE ED UNIONI

Capire il " C "

di Mr. Lambda

In questo incontro andiamo ad esplorare le strutture e le unioni, e naturalmente come utilizzarle nei nostri programmi in C.

Innanzitutto, perché utilizzare strutture? Le strutture vengono utilizzate per consolidare l'informazione relativa ad un certo ' oggetto '. L'oggetto in questione potrebbe essere una pagina in un' agenda di indirizzi, una registrazione in una lista di un inventario, una macchina in un parcheggio e qualsiasi oggetto che sia costituito da diversi pezzi di informazione che noi vogliamo tenere insieme in una singola entità.

In Pascal, questa unità viene chiamata un 'record'. In BASIC, voi potreste utilizzare l'istruzione FIELD per definire il complesso di informazioni che volete finiscano memorizzate sul disco. Queste parti di informazione vengono tenute in una forma tale per cui è facile accedervi ed è anche facile manipolarle. E non dimentichiamo che le strutture vengono utilizzate da molte chiamate di sistema di Amiga per gestire l'informazione di cui l'Amiga ha bisogno per controllare il suo sistema. L'informazione per creare window, per visualizzare sprite e bob, e per definire gadget è tutta contenuta in strutture. Voi dunque potete creare le vostre strutture nei vostri programmi per raggruppare efficientemente qualsiasi tipo di informazione.

Dichiariamo le nostre strutture

Immaginatevi una lista per la corrispondenza. Ciascuna persona nella lista ha l'informazione che la riguarda: il suo nome, cognome, indirizzo, CAP, città, prefisso e numero telefonico. In un linguaggio come il BASIC, ciascuno di questi elementi dovrebbe venire memorizzato in un array separato. Ciascun elemento dell'array corrisponderebbe a una certa informazione che riguarda la persona. In C, questa informazione può essere riunita in un singola struttura:

```
struct Registrazione
{
    char *Cognome;
    char *Nome;
    char *Indirizzo;
    char *CAP;
    char *Città;
    char *Prefisso;
    char *Telefono;
}
```

Il ' tag ' di questa dichiarazione di struttura è Registrazione. Per coloro che sono abituati a programmare in Pascal, tutto ciò dovrebbe sembrare familiare. L'equivalente in Pascal di una struttura viene chiamato un ' record '.

Dichiarare variabili della struttura Registrazione è davvero semplice, ecco qui:

```
struct Registrazione Persona1,
      Persona2, Agenda[500];
```

' Persona1 ' e ' Persona2 ' hanno entrambe sufficiente spazio riservato per contenere tutta l'informazione relativa a una registrazione nella nostra lista. ' DataBase ' è un array che può contenere fino a 500 diverse registrazioni del tipo presentato. Speriamo abbiate notato che i diversi elementi dell'array DataBase possono contenere solamente puntatori a stringhe di caratteri. L' array DataBase non contiene i valori delle stringhe! Ciascun elemento dell'array è una struttura del tipo ' Registrazione '. Ciascuna struct ' Registrazione ' contiene solamente puntatori a stringhe di caratteri.

Prendiamo a considerare l'esempio sotto, alla linea Persona1.Cognome = "Rossi". In questo esempio, l'elemento puntatore-a-caratteri ' Cognome ' viene inizializzato per puntare alla stringa "Rossi". In altre parole, il puntatore punta all'indirizzo della stringa "Rossi", che è una costante stringa compilata all'interno del programma. Naturalmente si può stampare questo valore utilizzando la funzione ' printf() '. Tuttavia, se voi desiderate solamente salvare questo database nel disco dovete farvi accorti, dal momento che ciascun elemento dell'array contiene solamente puntatori ai dati, non i dati stessi. Da questo punto di vista questo non è un pratico esempio di dichiarazione di un array per un semplice database.

Ci sono ulteriori complicazioni quando si utilizzano stringhe in C. Non è semplice come in BASIC. Ma, come al solito, il C ci offre una maggior flessibilità nel trattamento dei dati.

Ecco come si assegnano valori ai campi definiti precedentemente:

```
Persona1.Cognome = "Rossi";
Persona1.Nome = "Paolo";
Persona1.Indirizzo = "V. Dante,3";
Persona1.CAP = "00100";
Persona1.Città = "Roma";
Persona1.Prefisso = "06";
Persona1.Telefono = "73757";
```

In un certo senso, noi abbiamo appena dichiarato un nostro tipo di dati. Esso contiene tutte le informazioni che noi desideriamo contenere in una singola definizione. La parentesi graffa aperta e chiusa circoscrivono il contenuto della struttura. Ciascun elemento della struttura deve essere del tipo dichiarato: int, char, float, ... oppure di un altro tipo di struttura:

```
struct Conto
{
  char *Num_Conto;
  float Bilancio;
};

struct Registrazione_Bancaria
{
  char *Nome;
  struct Conto Credito;
  struct Conto Debito;
} Conti[3000];
```

Osservate che la struttura Conto non è una dichiarazione di variabile, ma essa è semplicemente una dichiarazione di un nuovo tipo di dati contenente un puntatore a caratteri e un numero in virgola mobile (float).

Le strutture non è però necessario che vengano dichiarate in questo ordine. Se la struttura Registrazione_Bancaria fosse stata dichiarata prima della struttura Conto, il compilatore assumerebbe che tale struttura sia definita successivamente nel programma. Quando voi utilizzate in una dichiarazione in C una struttura che non è stata ancora definita, ma che lo sarà in dichiarazioni successive, state utilizzando il cosiddetto ' forward reference ', riferimento in avanti.

E ora assegnare valori ai campi delle strutture è davvero semplice:

```
Conti[0].Credito.Bilancio = 12450000;
Conti[0].Debito.Bilancio = 14785000;
```

Uno dei vantaggi impliciti nell'utilizzo di strutture è che vi si può agire come fossero unità. Per esempio, invece di copiare ciascun campo individualmente da una struttura ad un' altra, voi potete copiare tutti i campi simultaneamente con un solo assegnamento:

```
Persona2 = Persona1;
```

Non è straordinario?!

Inizializzazione automatica

Il C vi consente di inizializzare automaticamente le variabili nel momento in cui le dichiarate. Per esempio, se voi dichiarate un intero e desiderate inizializzarlo a 500, ecco come potete fare:

```
int Numero = 500;
```

La stessa cosa può essere fatta con le strutture. Supponiamo che voi abbiate dichiarato una variabile di una struttura Registrazione, come presentato sopra, e che vogliate inizializzare ciascun campo ad un certo valore. Voi potete utilizzare un semplice statement come questo:

```
Persona1.Cognome = "Verdi";
Persona1.Nome = "Gerolamo";
Persona1.Indirizzo = "V. Po, 33";
Persona1.CAP = "00100";
Persona1.Città = "Roma";
Persona1.Prefisso = "06";
Persona1.Telefono = "55159";
```

Tuttavia, utilizzando l'inizializzazione automatica, la variabile potrebbe venire dichiarata in questo modo, ottenendo lo stesso risultato:

```
static struct Registrazione Persona1 =
{
    "Verdi";
    "Gerolamo";
    "V. Po, 33";
    "00100";
    "Roma";
    "06";
```

```
    "55159";
};
```

L'inizializzazione automatica può venire effettuata anche su array di strutture:

```
static
struct Registrazione DataBase[500] =
{
    { "Zero", "Zero", "Zero","Zero",
    "Zero", "Zero", "Zero"
    },

    { "Uno", "Uno", "Uno", "Uno",
    Uno", "Uno", "Uno"
            },
};
```

Questo inizializza i primi due elementi dell'array DataBase. A tutti gli elementi dell'array a cui non sia assegnato un valore esplicito viene assegnato automaticamente il valore zero.
Quando utilizzate l'inizializzazione automatica su strutture che contengono strutture, lo statement deve contenere le parentesi graffe intorno a ciascuna struttura:

```
static
struct Registrazione_Bancaria
            Persona3 =
{
   "Signor Bianchi",
   { "1590000", "254678500" },
   { "2567500", "256445500" }
};
```

Questo statement è equivalente a:

```
struct Registrazione_Bancaria
            Persona3;
Persona3.Nome = "Signor  Bianchi";
Persona3.Credito.Num_Conto = "1590000";
Persona3.Credito.Bilancio = "254678500";
Persona3.Debito.Num_Conto = "2567500";
Persona3.Debito.Bilancio = "256445500";
```

A questo punto è indispensabile introdurre un altro argomento importantissimo che riguarda tutti i tipi di dati da noi presentati, ma che presenta particolare interesse quando si vuole comprendere le strutture.

Allocare memoria per le strutture

I puntatori a strutture vengono dichiarati nello stesso modo in cui vengono dichiarati i puntatori a caratteri, oppure a interi:

```
struct Registrazione *Ptr;
```

Quando si utilizzano i puntatori alle strutture, per assegnare i valori desiderati a campi determinati, è necessario innanzitutto richiedere al sistema la memoria necessaria per l'allocazione della struttura. La chiamata di sistema per richiedere memoria al sistema è ' AllocMem() '. Ed eccovi un modo di utilizzare questa funzione per richiedere la memoria di cui avete bisogno:

```
Ptr = (struct Registrazione *)
AllocMem(sizeof(struct Registrazione),
                MEMF_PUBLIC);
```

Ora dobbiamo ammettere che la prima volta che abbiamo incontrato questo tipo di definizione eravamo non poco confusi. La funzione di chiamata deve avere due liste di parametri?! No, la prima parte dell'assegnamento, "(struct Registrazione *)" forza la chiamata di funzione AllocMem a restituirci un risultato che sia un puntatore del tipo struct Registrazione. Questo costrutto viene chiamato cast. In altre parole, il valore di ritorno della funzione AllocMem() è temporaneamente cambiato in un puntatore a Registrazione, in questo modo il compilatore non potrà lamentarsi che voi tentate di assegnare valori a variabili che sono in conflitto di tipo.

L'operatore sizeof

L'operatore ' sizeof ' nel primo argomento di AllocMem calcola la dimensione di una struttura Registrazione in byte. Sizeof può essere utilizzato con qualsiasi tipo dichiarato come argomento, e rappresenta un modo semplice di calcolare la dimensione dei diversi tipi di oggetti utilizzati. E soprattutto utile quando viene utilizzato con strutture molto ampie. E anche utile se voi volete scrivere del codice che sia davvero portabile. La dimensione dei tipi di dati può variare da macchina a macchina. Utilizzando l'operatore ' sizeof ' potete eliminare la dipendenza dalla macchina.

Naturalmente, voi potete decidere di sostituire l'intera espressione con il valore di ' sizeof ', ma perderete la portabilità. In questo caso infatti ci sono alcune gravi responsabilità che dovete assumervi: se voi in seguito deciderete di aggiungere un altro campo alla struttura che state utilizzando dovrete modificare quel numero in tutte le chiamate di AllocMem() presenti nel programma.

Attributi della memoria

Il secondo argomento di AllocMem() è MEMF_PUBLIC, e questo è uno dei quattro possibili attributi della memoria che descrivono le caratteristiche associate all'allocazione della memoria:
MEMF_PUBLIC: la memoria che non deve essere non indirizzabile. La memoria a cui possono riferirsi altri task deve essere pubblica, e questa caratteristica riguarda codice e dati.
MEMF_CHIP: la memoria che può essere utilizzata dai circuiti DMA. (Per esempio l'I/O del disco, le operazioni grafiche che utilizzano il Blitter, ecc.). La memoria MEMF_CHIP è circoscritta ai primi 512K di RAM.
MEMF_FAST: la memoria che è situata oltre i primi 512K di RAM. Questo tipo di memoria potrebbe venire utilizzato in operazioni critiche che non utilizzino i circuiti DMA.
MEMF_CLEAR: Questo attributo può essere utilizzato insieme con uno qualsiasi degli altri attributi.

La memoria che vi sarà restituita risulterà inizializzata a zero.
Voi potete effettuare l' ' or ' con il valore di MEMF_CLEAR per dire alla funzione AllocMem() di azzerare la memoria che state richiedendo.

E ritorniamo al nostro statement. A ' Ptr ' viene assegnato il valore di ritorno dalla chiamata a AllocMem().

Questo valore contiene l'indirizzo della prima word in memoria del blocco richiesto.
Gli assegnamenti delle strutture possono utilizzare anche puntatori in questo modo:

```
Ptr->Cognome = "Bianchi";
```

Osservate l'utilizzo di " -> " invece di " . " come nel precedente esempio. Per ora ci accontentiamo di avervi presentato anche questa possibilità, in un prossimo incontro ne tratteremo diffusamente.

Come le strutture vengono memorizzate

Quando le strutture vengono memorizzate nella memoria del computer, i loro campi vengono allocati in locazioni consecutive. Quando si allocano in memoria strutture come quella vista sopra, ci vengono restituite dal sistema come singoli blocchi di memoria. Quando vi riferite ad un campo in quella struttura, il C automaticamente fornisce un indice in quel blocco di memoria per individuare la locazione che voi desiderate utilizzare. E lavora analogamente per l'indicizzazione in un array. ' Ptr ' è l'indirizzo dell'inizio del blocco, e aggiungendo l'offset corretto, potete raggiungere la variabile a cui volevate riferirvi nella struttura.

Per esempio, quando viene invcontrata l'espressione "Ptr->Cognome", il C la converte in "Ptr+0", l'indirizzo di Ptr più l'offset di zero. Nello stesso modo, "Ptr->Città" viene convertito in "Ptr+16", cioè l'indirizzo di Ptr più un offset di 16.

I campi delle strutture sono ' allineati a word '. In altre parole, il C, invece di impaccare i campi nelle strutture per ridurre il più possibile le loro dimensioni, lascia dello spazio in alcuni campi. Se la dimensione di una variabile è più piccola di una word, il resto della parola è spazio inutilizzato. Per esempio la struttura:

```
struct a
{
  char a, b;
  int c;
  char d;
} esempio;
```

viene memorizzata in questo modo:

```
    byte   byte

  +--------------+
0 |  a  | vuoto |
  +--------------+
2 |  b  | vuoto |
  +--------------+
4 |     c  -----|
  +--------------+
6 |  d  | vuoto |
  +--------------+
```

L'allineamento è quindi una necessità per il C. Sappiate regolarvi il giorno in cui utilizzerete codice assembler insieme a codice C.

Unioni

Le unioni sono simili alle strutture, ma solamente un campo può essere presente nel medesimo istante. Da ciò si potrà arguire facilmente l'assoluta mancanza di qualsiasi carattere coniugale a cui il nome può far pensare. In altre parole, l'unione:

```
union
{
  int A;
  char B;
} temp;
```

può solamente allocare spazio per una variabile alla volta, sia per l'intero A, che per il carattere B. D'altra parte l'accesso ai singoli campi delle unioni richiede le stesse modalità viste per le strutture. Tutto questo potrebbe suonarvi strano o poco utile. Perché non dichiarare semplicemente una struttura ed avere un campo per A ed uno per B!?

Quando il C incontra un costrutto unione, esamina i campi all'interno di quell' unione per individuare il campo più ampio in quell'unione, e quindi riserva la quantità di spazio ad esso relativo. Normalmente le unioni vengono utilizzate quando si presenta la necessità di memorizzare solamente un tipo di variabile alla volta nella nostra struttura dati, e quindi abbiamo bisogno - oppure desideriamo - di risparmiare spazio. In Pascal, l'equivalente delle unioni sono i ' record variabili '. Ma vediamone subito un esempio.

In un inventario per un magazzino alcuni articoli vengono stimati in denaro (scatolette, elettrodomestici, ecc.) e altri vengono misurati in base alla quantità presente in magazzino (verdura, frutta, ecc.). Per descrivere questo nostro tipo di organizzazione degli articoli presenti potremmo utilizzare questa struttura:

```
struct Articolo
{
      char *Nome;
      int Lire;
      float Quantità;
};
```

e in relazione al valore assunto dal campo Nome, il programma potrebbe riferirsi a Lire oppure a Quantità. In alternativa però, potremmo utilizzare la seguente struttura:

```
struct Articolo
{
  char *Nome;
  union
  {
    int Lire;
    float Quantità;
  } num;
};
```

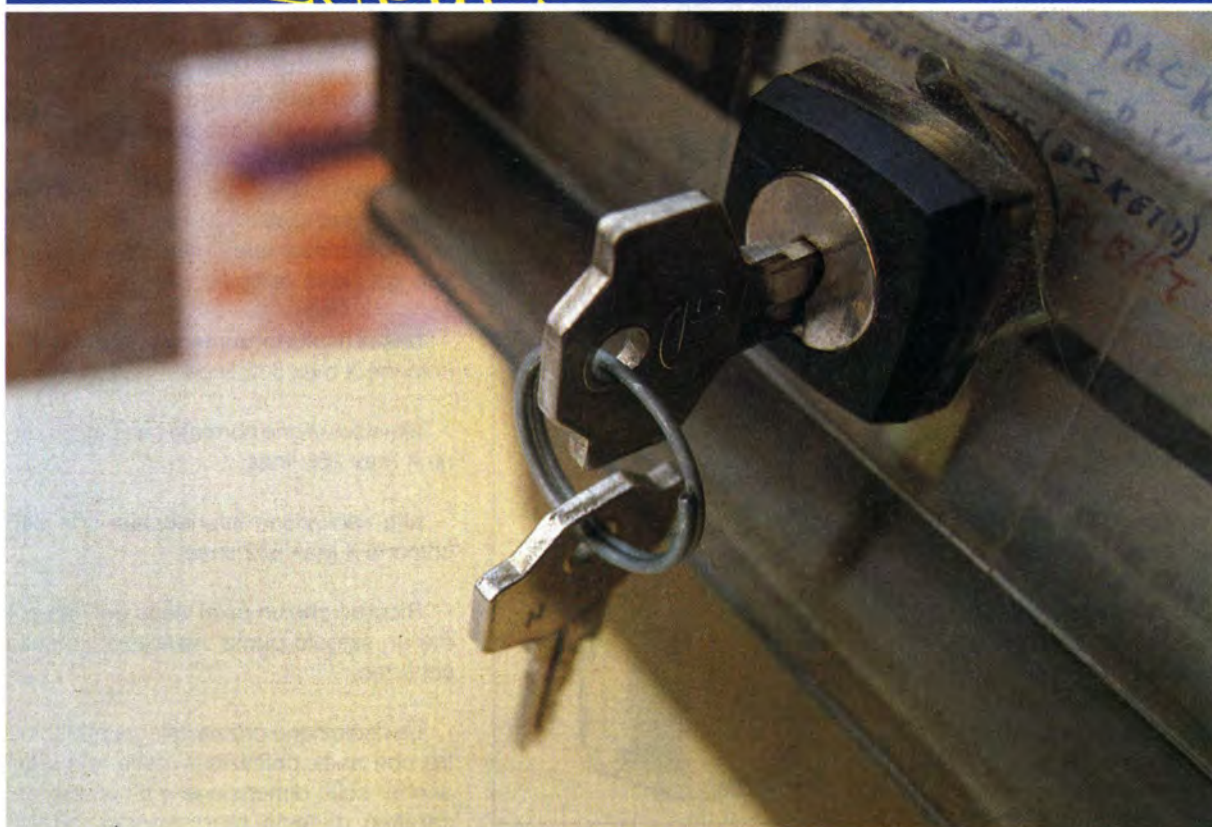
Come potete osservare siamo ancora in grado di determinare a quale campo dell'unione dobbiamo riferirci in dipendenza del valore assunto dal campo Nome, ma abbiamo risparmiato dello spazio in memoria per l'allocazione della struttura. In un'applicazione in grande scala, l'utilizzo intelligente delle unioni può far risparmiare dello spazio considerevole di memoria.

E proprio grazie al modo in cui vengono allocate le unioni in memoria che in C potete effettuare operazioni che normalmente non potete fare. Per esempio, dovete scrivere una funzione per convertire numeri in virgola mobile dal modo in cui Amiga li rappresenta internamente al modo in cui li rappresenta un altro tipo di computer. Per realizzare questa funzione potreste dover manipolare bit, ma in C è illegale manipolare bit con i numeri in virgola mobile. Il compilatore semplicemente non lo farà. Tuttavia c'è un piccolo trucco che si serve proprio delle unioni:

```
union
{
  float a;
  int b[2];
  char c[4];
} D;
```

Ingegnoso non è vero !? Dopo aver assegnato il numero da scomporre a D.a, possiamo vedere in D.b[0] e D.b[1] l'ordine delle word bassa e alta del numero, e in D.c[0], D.c[1], D.c[2] e D.c[3] persino i singoli byte che lo compongono. E per questa volta vi abbiamo detto anche troppo; fateci sapere se ne avrete saputo trarre durevole giovamento. Al prossimo incontro.

LA GESTIONE DI INTUITION

SVILUPPIAMO UNA LIBRERIA DI FUNZIONI PER GESTIRE INTUITION

di Antonio Ferrillo

Così come è facile e intuitivo l'approccio all'Amiga, anche da parte di chi non è avvezzo all'utilizzo del computer, altrettanto è difficoltosa e non immediata la comprensione della filosofia del sotfware di base responsabile di un cosi piacevole approccio.

L'ambiente software

Parlare di Sistema Operativo come è tradizionalmente inteso è piuttosto riduttivo nel caso dell'Amiga, in realtà questa macchina è dotata di un complesso di sottosistemi, ognuno con un ruolo ben definito, organizzati in un insieme gerarchico strutturato su diversi livelli logici in relazione al controllo dell'Hardware.

In cima alla struttura vi è l'AmigaDOS che è quel sottosistema che più si avvicina ad un Sistema Operativo propriamente inteso, è quello che si attiva per primo al momento dell'avvio della macchina presentandosi in una sua veste peculiare : quella dell'organizzatore ed archiviatore dei dati di qualunque tipo che devono risiedere su memoria esterna.

Subito dopo mette a nostra disposizione un'altra sua funzione: quella di interfaccia tra utente e sistema, infatti ci presenta una finestra sul display invitandoci al colloquio via Console tramite i Comandi CLI oppure è pronto ad avviare un processo elaborativo se digitiamo il nome di un programma al Prompt.

Ma il più delle volte questa maniera di colloquiare così simile a quella offerta da un computer tradizionale non ci soddisfa; così facciamo andare il Boot avanti fino al caricamento del Workbench,il sistema di interfaccia alternativo, con le sue finestre, le icone, i menu, il mouse.

Più limitato rispetto all'AmigaDOS ma più immediato da utilizzare, esso sfrutta i servigi dell'AmigaDOS nell'espletamento delle sue funzioni ed ambedue interagiscono con i sistemi situati ai livelli inferiori e questi fra di loro in un intreccio complesso che rende questa macchina tanto poliedrica quanto difficile da capire.

Ai livelli sottostanti troviamo Intuition con i suoi oggetti: menus, gadgets, requesters, alerts e via dicendo orientati a facilitare il colloquio uomo-macchina.

Vi sono ancora i Devices, che presiedono all'accesso alle periferiche e all'Hardware sonoro, le Librerie Grafiche e dei Piani con le loro routines per produrre i diplays, ed inoltre l'Exec, che ha il delicato compito della gestione del Multitasking disciplinando l'accesso alla CPU da parte dei Tasks in attività.

La programmazione di Amiga in C

Chi si avvicina al mondo della programmazione di Amiga col C ha la possibilità di accedere alle componenti del sistema sostanzialmente secondo due livelli di sofisticazione rispetto all'Hardware.

Al livello più alto si rimane in ambiente AmigaDOS sfruttando le routines standards del C che permettono di accedere alle risorse tradizionali di ogni DOS cioè il Console I/O, il Disk drive I/O, la Stampante, etc. il programma che ne risulta è notevolmente portabile ma il campo di applicazione rimane ristretto a quelle di tipo amministrativo-contabile tipiche dei Personal tradizionali che l'Amiga comunque è in grado di supportare egregiamente.

Al livello più basso entriamo nel mondo Amiga, infatti possiamo accedere direttamente alle routines dell'AmigaDOS senza il filtro delle funzioni del C, accediamo alle periferiche tramite i Devices, entriamo nell'angolo affascinante della grafica e di Intuition sfruttando le routines residenti nelle librerie relative.

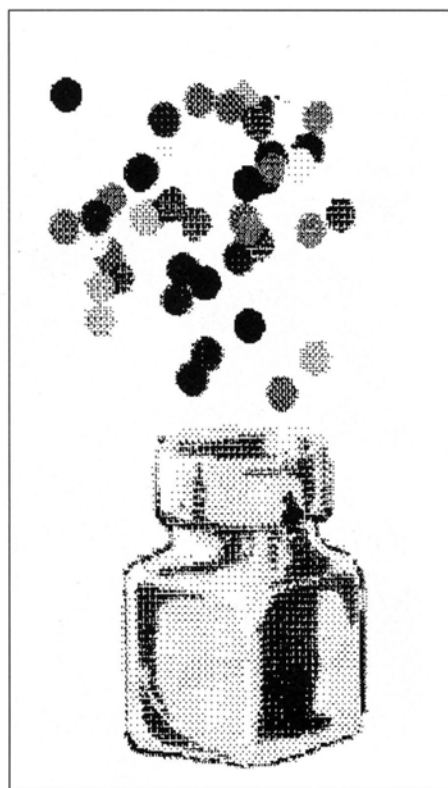
Infine ci serviamo delle routines di Exec per condizionare l'elaborazione del nostro Task-programma e per colloquiare con gli altri Tasks contemporaneamente attivi.

Più giù ancora possiamo controllare direttamente i processori grafici e sonori per applicazioni particolarmente spinte che sfruttino a fondo le risorse hardware, ma per queste, forse è più indicato il linguaggio Assembler.

Cosa vogliamo fare

L'obiettivo che ci prefiggiamo è quello di programmare l'Amiga a basso livello col C iniziando da Intuition e utilizzando un metodo di realizzazione dei programmi basato sullo sviluppo di librerie private di funzioni Amiga da affiancare a quelle fornite dal compilatore usato.

I vantaggi che individuiamo in tale sistema sono essenzialmente :

1 - Minor numero di linee di codice in un programma e quindi minor fatica di digitazione e minore possibilità di errori.

2 - Maggior pulizia logica e quindi migliore leggibilità

3 - Maggior produttività per la possibilità di dedicare le proprie risorse principalmente alla soluzione del problema applicativo.

Intuition

La parola Intuition identifica quell'insieme di routines catalogate nella intuition.library residente in ROM e che sono a disposizione del programmatore per facilitare e rendere appunto intuitiva l'interazione tra l'utente e il programma in elaborazione.

Schermi

Lo schermo è la componente primaria di Intuition, quella che serve per definire le caratteristiche dell'ambiente di display di tutti gli altri oggetti che vi compariranno sopra.

Gli elementi che lo caratterizzano sono principalmente le dimensioni, la risoluzione, e il numero dei colori visualizzabili, tutti interdipendenti tra loro.

I modi di risoluzione sono quattro :

bassa risoluzione normale che permette di displayare 320 colonne di pixels sul piano orizzontale X max 256 linee di pixels sul piano verticale.

bassa risoluzione interlacciata con 320 colonne X max 512 linee.

alta risoluzione normale con 640 colonne X max 256 linee.

alta risoluzione interlacciata con 640 colonne X max 512 linee.

Ricordo che un pixel viene definito come un singolo punto visualizzabile sullo schermo.

La risoluzione orizzontale ha effetto oltre che sulla definizione delle immagini anche sulla dimensione e il numero dei caratteri di testo rappresentati, questo perché sull'Amiga non c'è uno speciale modo testo, ma il testo stesso è disegnato sullo schermo grafico.

Ad esempio usando la Font di default Topaz 8 residente in ROM, è possibile visualizzare in alta risoluzione 80 caratteri per linea che in bassa risoluzione diventano 40, come sul vecchio 64.

La risoluzione verticale interlacciata ha gli stessi effetti sulla definizione delle immagini e sul numero e dimensione dei caratteri rappresentabili, ma il modo interlacciato è caratterizzato dal fastidioso fenomeno del 'flickering' che tutti avrete sperimentato.

In realtà nel modo interlacciato il raddoppio delle linee viene ottenuto con la visualizzazione di un secondo display sfasato di una linea in basso rispetto al primo, ma questo avviene nello stesso tempo impiegato per un unico display (1/60 di secondo).

Perciò nel modo interlacciato ogni schermata permane sul video per un tempo che è la metà rispetto al modo normale (1/120 di secondo) : troppo breve perché l'immagine permanga a sufficienza sulla nostra retina.

Per quanto riguarda le dimensioni, per noi è possibile gestire solo l'altezza, lo schermo infatti si estende sempre a partire dalla colonna 0 rispetto al margine sinistro ed è largo quanto permesso dalla risoluzione orizzontale prescelta.

E' possibile invece determinare l'altezza e la distanza dal top del display, dal che si deduce che è possibile aprire più di uno schermo sul display anche con risoluzione differente, ma solo sul piano verticale.

Il numero dei colori possibili è determinato dal numero di bit planes associati allo schermo : un bit plane, letteralmente piano di bit, è un'area di memoria in cui ogni bit contiene l'informazione riguardante il colore di un pixel.

Poiché un bit può rappresentare 2 valori : 0 e 1, corrispondenti a due colori possibili, se noi sovrapponiamo diversi bit planes incrementiamo le possibilità di scelta di un fattore 2 per ogni bit plane aggiunto.

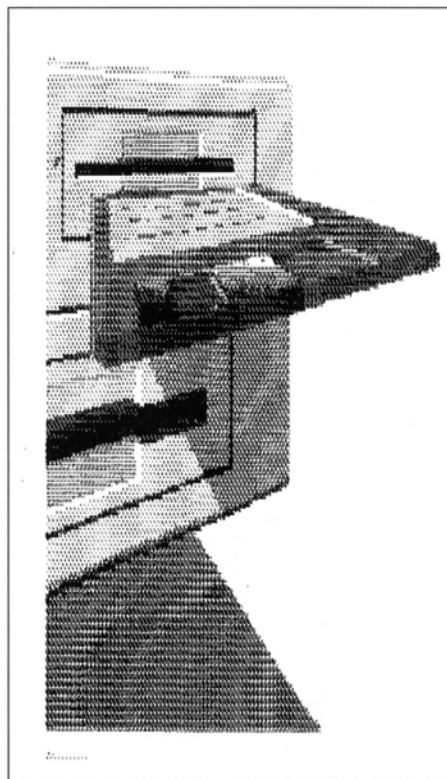
Il numero massimo di bit planes associabili allo schermo è determinato dalla risoluzione orizzontale prescelta : in hi res questo numero è 4 per un totale di 2 elevato alla quarta colori cioè 16; in low res è 5 per un totale di 32 colori.

La configurazione binaria ottenuta dai bits sovrapposti corrisponde ad un numero che è quello del registro colore che contiene le informazioni per l'hardware che pilota il display.

Il Workbench

Il sistema ci mette a disposizione il suo schermo di default, detto del Workbench ma in realtà condiviso anche dall'AmigaDOS che vi apre la sua finestra, si tratta di uno schermo ad alta risoluzione normale con 2 bit planes associati quindi 4 colori possibili: se queste caratteristiche ci soddisfano possiamo fare a meno di aprire uno schermo nostro risparmiando la fatica e la memoria necessaria.

Come aprire uno schermo

Creare uno schermo è una cosa piuttosto semplice : occorre aprire innanzitutto la intuition.library per poter accedere alle routines in essa contenute mediante una chiamata alla routine dell'Exec OpenLibrary() la quale ci ritorna l'indirizzo della libreria stessa che utilizzeremo per chiuderla a fine lavoro; ricordo semplicisticamente che la exec.library, con la dos.library, è l'unica libreria che è sempre aperta e che quindi non necessita di apertura per accedere alle sue routine.

Questa apertura ci permetterà di chiamare la OpenScreen(), ma prima dobbiamo inizializzare una struttura di tipo NewScreen predefinita nel file include <intuition/intuition.h> e riportata in fig. 1.

Settando questa struttura informiamo Intuition sulle caratteristiche che dovrà avere il nostro schermo : determiniamo la posizione rispetto al top del display in pixels, l'altezza, sempre in pixels; ricordo che la posizione rispetto al left side è sempre 0 e che la larghezza è uguale a 320 o 640 in funzione della risoluzione prescelta.

Nel campo Depth va il numero di bit planes, quindi settiamo il registro colore dei tratti e del titolo, il registro colore della barra, ancora vi associamo una Font di caratteri, ma se vogliamo utilizzare quella di default inseriamo un NULL.

Continuiamo inserendo un pointer ad una stringa contenente il titolo, mettiamo un NULL nel pointer Gadgets in quanto attualmente non utilizzato, infine, se vogliamo disegnare noi una mappa di memoria da utilizzare in sostituzione della display memory di Intuition, in CustomBitMap ne inseriremo l'indirizzo.

Fillata la struttura NewScreen chiameremo la OpenScreen() la quale ci restituirà il puntatore ad una struttura di tipo Screen nella quale Intuition oltre a ricopiare alcuni campi prelevati dalla NewScreen avrà inserito delle informazioni di cui alcune potranno essere utilizzate nel nostro programma e delle quali parleremo quando sarà necessario.

Se l'apertura è andata male, magari per mancanza di memoria o perché qualche dato inserito nella NewScreen non è esatto, il pointer alla struttura Screen sarà impostato a NULL.

Come vedete il colloquio con Intuition avviene utilizzando blocchi di memoria inizializzata secondo il formato di strutture predefinite : noi passiamo una struttura a lui ed egli ne ritorna una a noi.

Finestre

Le finestre sono delle zone di schermo che un programma si riserva per visualizzarvi i suoi outputs e per accettare i suoi inputs.

Esse rappresentano un vero e proprio terminale virtuale a disposizione dell'applicazione la quale può aprirne anche più di una sullo stesso schermo così come possono coesistere finestre aperte da Tasks diversi che girano in contemporanea e che magari colloquiano fra di loro.

In un dato momento solo una finestra può essere attiva sullo schermo, cioè può accettare degli input,le altre continuano a ricevere i loro output, ma sono inibite a ricevere input fino a che l'utente non vi clicka dentro col mouse, l'abilitazione all'attivazione viene comandata da program-

ma settando un flag opportuno.

Un programma può ricevere gli input via finestra attiva in due modi: tramite IDCMPs cioè l'Intuition Direct Communications Message Ports, di cui parleremo prossimamente, che gli trasmette gli eventi esterni generati dalla tastiera, dal mouse, o da altra periferica abilitata; oppure tramite Console Device che gli fornisce i caratteri digitati sulla tastiera già convertiti in formato ASCII e quindi immediatamente utilizzabili.

Il programma stesso può inviare i suoi output grafici alla finestra sfruttando alcune routines grafiche primitive; può inviare altresì i testi tramite alcune routin grafiche di testo, oppure via Console Device, o anche può usare alcune routines messe a disposizione da Intuition stesso.

E' possibile associare alla finestra molti oggetti di Intuition come menus, gadgets, requesters, etc. i quali costituiscono anche un modo ulteriore di acquisire inputs.

Tramite i gadgets di sistema una finestra può essere rimpicciolita, ingrandita, spostata sullo schermo, può essere fatta sparire dietro le quinte o riportata in primo piano, può essere infine chiusa dall'utente: questo è in genere il segnale che l'utente stesso vuole chiudere l'elaborazione,e il programma, intercettando il messaggio di chiusura farà le opportune operazioni di cleanup tra le quali la chiusura stessa della finestra.

Vi sono vari tipi di finestre visualizzabili

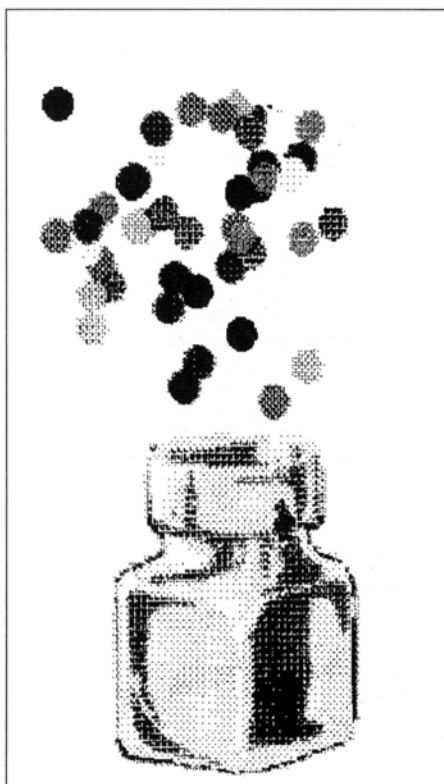
BACKDROP : si tratta di una finestra attiva nascosta sul retro dello schermo.

BORDERLESS : è una finestra senza bordi, in combinazione con BACKDROP permette di avere il display completamente libero per rendere immagini.

GIMMEZEROZERO : il suo nome deriva dal fatto che le coordinate relative al top ed al left side dello schermo sono sempre 0,0, in realtà per il display di questo tipo di finestra sono utilizzate due superfici, una esterna, sulla quale viene contornata la finestra, ed una interna che costituisce l'area utilizzabile e le cui coordinate vengono memorizzate da Intuition in alcune variabili cui ci si può riferire per restare entro i limiti quando vi si scrive o vi si disegna dentro.

SUPERBITMAP : è un tipo di finestra per la quale invece di utilizzare la display memory allocata da Intuition ne creiamo una noi per creare magari una finestra personalizzata.

Nel definire una finestra noi dobbiamo anche stabilire cosa accade al contenuto quando esso viene ricoperto sullo schermo da un altro oggetto.

Nel modo SIMPLE_REFRESH il contenuto oscurato viene perso e quindi è a carico del programma rivisualizzarlo.

Nel modo SMART_REFRESH Intuition mantiene in memoria una copia del diplay quindi il contenuto viene ripristinato quando sparisce l'oggetto oscurante.

Nel modo SUPERBITMAP la display memory, essendo un'area privata allocata dal programma, viene sempre salvata.

Apertura di una finestra

Per aprire una finestra dobbiamo innanzitutto aprire la intuition.library, se non l'abbiamo già fatto per aprire uno schermo nostro, quindi dobbiamo fillare una struttura di tipo NewWindow che si trova nel file include <intuition/intuition.h> e che riportiamo in fig. 2.

Nel riempire questa struttura informiamo Intuition sulle caratteristiche che deve avere la nostra finestra; indichiamo infatti la posizione rispetto al top ed al left side dello schermo, l'altezza e larghezza in pixels,il colore dei tratti e di fondo della barra orizzontale.

Nella variabile Flags settiamo i bits relativi all'attivazione o meno della finestra, al modo di salvataggio del contenuto quando viene ricoperto, alle possibilità di manipolazione da parte dell'utente esterno.

Nella variabile IDCMPFlags settiamo invece i bit relativi ai tipi di messaggi che vogliamo ricevere tramite IDCMPs, questi bits, cosï come quelli relativi al campo Flags sono definiti nel file

<intuition/intuition.h>

con un nome simbolico nel corpo di una lunga voce in cui quel bit è in ON, essi vengono trasferiti nella variabile di destinazione mediante una operazione di OR.

Proseguendo nell'inizializzazione informiamo Intuition sul tipo di schermo cui associare la finestra, se si tratta di un CUSTOMSCREEN, cioè di uno schermo nostro, nel campo Screen inseriremo il pointer alla struttura di tipo Screen ritornatoci dalla OpenScreen(), altrimenti NULL se si tratta di un WBENCHSCREEN.

Se abbiamo dei gadgets da associare alla finestra, nel pointer FirstGadget metteremo l'indirizzo della struttura di tipo Gadget che definisce il primo della lista, altrimenti NULL : i gadgets possono sempre essere associati in un secondo momento.

Nel pointer CheckMark metteremo l'indirizzo di una struttura di tipo Image se vogliamo far apparire una immagine sottostante l'item di un menù quando viene selezionato, altrimenti NULL.

Per una window col bit SUPERBITMAP in ON nel campo Flags, metteremo l'indirizzo della struttura di tipo BitMap, che definisce la nostra display memory, nel pointer BitMap.

Infine chiudiamo con le dimensioni minime e massime che la finestra può avere quando viene manipolata dall'utente.

Settata la struttura NewWindow ne pas-

siamo l'indirizzo alla OpenWindow() la quale ci ritornerà l'indirizzo di una struttura Window che intuition avrà preparato per noi.

Se qualcosa è andato storto, in quell'indirizzo ci sarà NULL oppure avremo l'ammutolimento completo del sistema, in quel caso bisognerà verificare l'esattezza dell'inizializzazione della struttura NewWindow, controllare che la posizione e le dimensioni siano entrocontenute nello schermo sia esso custom o workbench, che i registri colore scelti siano in accordo coi bit planes associati allo schermo e così via, se tutto è a posto pensate ad una mancanza di memoria disponibile.

La struttura Window che ci ritorna la OpenWindow() oltre a contenere notizie ricopiate dalla NewWindow ne riporta altre utilizzabili nel nostro programma, di cui si presenterà senz'altro l'occasione di parlare.

Le nostre funzioni

Allo scopo di dimostrare praticamente come si gestiscono gli schermi e le finestre ho preparato due programmi contenenti funzioni di gestione di schermi e finestre di impiego generalizzato.

Troverete i sorgenti anche sul dischetto accluso, se avete un compilatore Lattice o Aztec potrete compilarli senza modifiche, infatti essi sono stati scritti in un codice compatibile con i due compilatori C per Amiga più accreditati : il Lattice nella versione 3.10 e il Manx Aztec nella versione 3.4.

Dopo averli compilati vi consiglio di inserire i moduli oggetto in libreria privata, più avanti vi spiegherò come fare.

Essi sono i primi di una serie che vi permetterà alla fine di disporre di uno ottimo strumento di sviluppo di programmi per Amiga.

Se non volete farvi una libreria privata potrete conservarli sotto forma di sorgente,e quando necessario, adattarli alle esigenze del programma chiamante e linkarli assieme ad esso dopo averli compilati.

La propensione del C alle funzioni è tale che con pochissima fatica potrete anche incorporarli nel programma principale : appendeteli al sorgente principale con un buon editor o col comando join della CLI, trasferite le dichiarative esterne nel principale prima della main() avendo cura di non duplicare le #include ed il gioco è fatto.

fig. 1

```
struct NewScreen
{
        SHORT LeftEdge, TopEdge;
        SHORT Width, Height, Depth;
        UBYTE DetailPen, BlockPen;
        USHORT ViewModes;
        USHORT Type;
        struct TextAttr *Font;
        UBYTE *DefaultTitle;
        struct Gadget *Gadgets;
        struct BitMap *CustomBitMap;
};
```

Sreen.c

Il programma screen.c contiene le funzioni di apertura di schermo, attivazione,chiusura e ritorno dell'indirizzo della struct Screen.

Quest'ultima funzione è stata prevista allo scopo di liberare il chiamante dall'impegno di mantenere quest'indirizzo.

Il listato si commenta da sè, ricordo solo che è possibile aprire 2 schermi contemporaneamente e che la routine Screen_Front() permette di portare sul davanti del display uno schermo precedentemente relegato sul retro.

fig. 2

```
struct NewWindow
{
        SHORT LeftEdge, TopEdge;
        SHORT Width, Height;
        UBYTE DetailPen, BlockPen;
        ULONG IDCMPFlags;
        ULONG Flags;
        struct Gadget *FirstGadget;
        struct Image *CheckMark;
        UBYTE *Title;
        struct Screen *Screen;
        struct BitMap *BitMap;
        SHORT MinWidth, MinHeight;
        USHORT MaxWidth, MaxHeight;
        USHORT Type;
};
```

Window.c

Il programma window.c è strutturato in maniera simile allo screen.c, esso permette di aprire fino a 4 finestre contemporaneamente, le funzioni sono quelle di apertura, chiusura, ritorno dell'indirizzo della struct Window.

Anche se vengono effettuati alcuni controlli sugli argomenti ricevuti ricordo di porre molta attenzione all'esattezza degli argomenti stessi i quali devono essere congruenti fra di loro e in accordo con l'ambiente di schermo creato prima.

Il programma dimostrativo

Per dimostrare l'utilizzo delle routines di cui sopra ho preparato un programmino dimostrativo che troverete pure sul dischetto accluso anche in versione eseguibile.

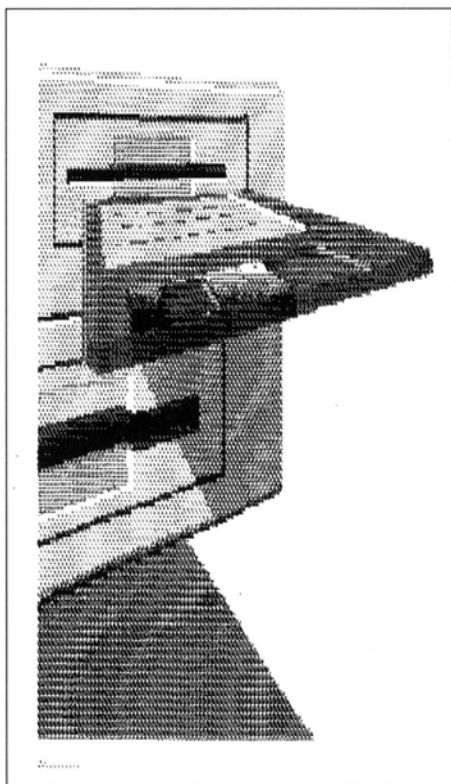
Esso pretende di essere anche un esempio di come usare le funzioni interne e l'indentazione per rendere piu' leggibile il codice.

E' ampiamente commentato e contiene qualche anticipazione di argomenti che verranno trattati in futuro, come per esempio la gestione dei messaggi di Intuition.

Compare anche un altro oggetto di Intuition : l'Alert, quel rettangolo rosso lampeggiante che compare drammaticamente in alto sullo schermo per annunciarvi che il sistema è morto e non c'è altro da fare che resettare.

Qui è stato usato in luogo della solita printf() per avvertire che c'è stato qualche problema durante un'apertura e che l'elaborazione non può proseguire, anzi, con la certezza che andrà tutto bene durante l'esecuzione ne è stato inserito uno apposta per farvelo vedere.

Si tratta di un Alert di tipo RECOVERY: si usa per indicare che è andato a pallino il programma ma che il sistema è ancora in piedi.

Infine, per rendere meno noioso il tutto, è stata usata una routine grafica di base per cambiare il colore dello schermo in accordo coi movimenti del mouse.

Lo sviluppo di un programma in C

Il C è un linguaggio compilato, vuol dire che un modulo scritto in C ha bisogno di subire un processo di compilazione per tradurre le istruzioni in linguaggio macchina trasformandolo in modulo oggetto (.o), tuttavia esso non è ancora un programma eseguibile, per diventarlo deve subire ancora un processo detto di linkage editor allo scopo di incorporare nel programma le routines esterne e stabilire gli opportuni legami.

Le routines esterne possono risiedere sia in libreria di compilatore che in libreria privata, sia in altri moduli oggetto precompilati ed associati al principale nel passo di link.

L'output generato è finalmente un programma pronto per essere eseguito.

I passi di compilazione e di link produrranno una diagnostica degli errori e dei warnings, gli errori dovranno essere corretti prima di risottoporre i moduli, i warnings, anche se non inficiano il risultato della compilazione, vanno comunque esaminati, in quanto potrebbero essere la spia di anomalie in sede di esecuzione.

Se venite dal Basic e siete alle prime armi col C dovrete faticare parecchio prima di riuscire a far girare un programma degno di questo nome, ma non è il caso di scoraggiarsi perché è normale che il C richieda una certa prima fase di assuefazione.

Catalogazione in libreria privata

Per catalogare un modulo in libreria privata occorre utilizzare una apposita utility.

I possessori del Lattice troveranno sul dischetto accluso una utility per la gestione di una libreria privata compatibile col Blink, essa si chiama Alib ed è tratta dal volume 2 della raccolta di software di pubblico dominio di Fred Fish, l'Alib deve essere ricopiata nella directory c del volume 1 del compilatore.

I possessori dell'Aztec utilizzeranno invece l'utility lb gia' fornita a corredo del compilatore.

Direttive di compilazione

Per i lettori che fossero impazienti di sviluppare il loro primo programma prima di aver digerito il manuale del compilatore in loro possesso riporto le linee di comando più essenziali per portare a buon fine il lavoro, senza pretendere di sostituirmi al manuale che consiglio di leggere attentamente.

Assumiamo che il programma principale si chiami demo.c e che il modulo contenente le funzioni si chiami screen.c; la libreria privata la chiamiamo p.lib.

Comandi Lattice

Per compilare un sorgente : **lc -v demo.c lc -v screen.c**
Per catalogare in libreria privata : **alib r lib/p.lib screen.o se la libreria non esiste verrà inviato un messaggio a video con la richiesta di autorizzazione a crearla : rispondete y.**
Per deletare un modulo da libreria privata : **alib d lib/p.lib screen.o**
Per rimpiazzare un modulo con la versione aggiornata : **alib r lib/p.lib vecchioscreen.o nuovoscreen.o**
Per listare il contenuto di una libreria : **alib l lib/p.lib**
Per linkare un modulo : **blink from lib/c.o+demo.o library lib/p.lib+ lib/lc.lib+lib/amiga.lib nodebug smallcode**
Per linkare più moduli : **blink from lib/c.o+demo.o+screen.o library lib/lc.lib+lib/amiga.lib nodebug smallcode**

Comandi Aztec

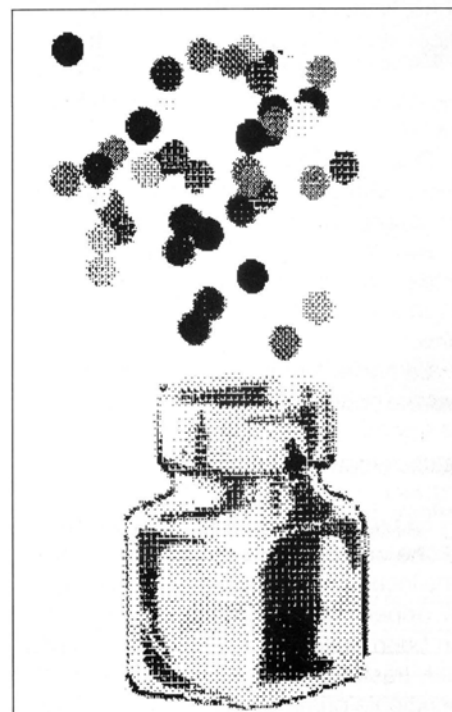
Per compilare un sorgente : **cc demo.c -s cc screen.c -s**
Per catalogare un modulo in libreria privata quando questa non esiste :**lb lib/p.lib screen.o**
Per catalogare un modulo quando la libreria esiste già : **lb lib/p.lib -a+ screen.o**
Per deletare un modulo da libreria : **lb lib/p.lib -r screen.o**
Per rimpiazzare un modulo con la versione aggiornata:**lb lib/p.lib -r vecchioscreen.o nuovoscreen.o**
Per spostare un modulo prima di un altro all'interno di una libreria : **lb lib/p.lib -b primadi.o screen.o**
Per listare il contenuto di una libreria : **lb lib/p.lib -t**
Per linkare un modulo : **ln demo.o p.lib c.lib m.lib oppure : ln demo.o -lp -lc -lm**
Per linkare più moduli : **ln demo.o screen.o -lc -lm**

Integrate eventualmente queste linee con il path corretto (root:dir/) in funzione di come avete installato il vostro compilatore per evitare messaggi tipo not found e simili. Buon lavoro.

SoftMail

Vendita per corrispondenza di programmi originali ed accessori per computers.

VIA NAPOLEONA 16 - 22100 COMO - TEL. (031) 30.01.74

® SoftMail è un marchio registrato da Lago snc

Presentiamo in questa pagina alcune tra le ultime novità disponibili dal catalogo SoftMail. Ecco alcune informazioni utili per utilizzare il servizio SoftMail: è possibile effettuare ordini telefonicamente SOLO se è già stata effettuata una spedizione a proprio nome ed è stata regolarmente ritirata. Dal secondo in poi accettiamo ordini telefonici. Chi desidera notizie sulla disponibilità ed i prezzi dei prodotti che non compaiono in questa lista può chiamare lo (031)30.01.74 dalle 14:30 alle 18:00. SoftMail può organizzare la consegna anche tramite corriere: interpellateci per maggiori informazioni. Oltre alle ultime novità qui esposte, SoftMail offre l'intero catalogo delle seguenti case: Aegis, Cinemaware, EA, Microprose, Rainbird, SSI, SSG, Sublogic.

ACCESSORI

Copritastiera A500	25.000
Final cartdrige III	110.000
Flicker master	29.000
Joy. Slik stick	16.500
Joy. SpeedKing	29.000
Joy. SpeedKing Autof.	33.000
Joy. Tac 2	29.000
Joy. Tac 5	39.000
MouseMat tappetino	22.500
Coprimouse	20.000
Portamouse	12.500
Portadischi 3" (30)	34.000
Portadischi 5" (40)	37.000
SlimLine (tastiera 64)	49.000
Voice messenger C64	60.000

LIBRI/HINTS & TIPS

Alternate city clue specificare 8/16 bit	18.000
Bard's tale I clue	22.500
Bard's tale II clue	25.000
Bard's tale III clue	25.000
Black cauldron clue	18.000
Deathlord clue	telef.
Dungeon master clue	25.000
Elite clue	18.000
Graphic adv.creator hint	7.500
Mars saga clue	telef.
Might & Magic clue	25.000
Pool of radiance cle	20.000
Quest for clues	39.000
Sentinel world clue	25.000
Starflight clue	25.000
Ultima V clue	22.500
Wasteland clue	16.500

AMIGA

Heroes lance hints disk	*telef.*
Bard's tale hints disk	*telef.*
Aegis draw 2000	telef.
Aegis images	69.000
Aegis impact	125.000
Aegis sonix	110.000
Aegis videotitiler 1.1	185.000
Alternate reality: the city	69.000
Animation multiplane	125.000
Armageddon man	49.000
Audiomaster	85.000
Audiomaster II	telef.
Baal	29.000
Barbarian II	telef.
Batman	29.000
Battlechess	49.000
Black cauldron	9.000
Blitzkrieg at Ardennes	59.000
California games	25.000
Carrier command	49.000
Cell animator	telef.
Chrono quest	65.000
Comic setter	139.000
Comic setter art disks	telef.
Corruption	49.000
Cosmic pirates	telef.
Crazy cars	29.000
Def con 5	59.000
Defender of the crown	59.000
Deluxe print II	99.000
DigiPaint	99.000
DigiView GOLD	telef.
Director	99.000
Disk drive esterno	299.000
Dos2Dos	75.000
Dragon's lair (1 MByte, 6 disks)	75.000
Driller TUTTO italiano	59.000
Dungeon master (1 MB)	59.000
Elite	45.000
Empire	49.000
F16 Falcon	59.000
Fantavision	125.000
Fire & forget	39.000
Fish	45.000
Flight simulator II	99.000
Scenery disks	telef.
Galileo 2.0	99.000
Garfield	49.000
Gauntlet II	25.000
Grabbit	49.000
Hellfire attack	39.000
International soccer	39.000
Italy 90 soccer	39.000
Incr.s.sphere	49.000
Jet & Japan bundle	110.000
Joan of Arc	25.000
King of Chicago	49.000
LED storm	25.000
Legend of the sword	45.000
Life cycles vol.1	39.000
Light!Camera!Action!	99.000
Lords of the Rising Sun	telef.
Major motion	39.000
Maxiplan plus	299.000
Menace	39.000
Mickey Mouse	25.000
Modeler 3D	129.000
Murder on Atlantic	59.000
Nigel Mansell grand prix	49.000
Obliterator	45.000
Offshore warrior	39.000
Outrun	25.000
P.O.W.	59.000
PacMania	25.000
Phantom fighter	45.000
Pioneer plague	59.000
President is missing	49.000
ProSound designer	79.000
ProSound w/hardware	199.000
Prowrite 2.0	179.000
Publisher plus	125.000
Reach for the stars	59.000
Rebel...Chickamauga	telef.
Return to Genesis	39.000
Rocket Ranger	59.000
Roger Rabbit	69.000
SEUCK	telef.
Sex vixens in space	55.000
Sculpt/Animate 4D	telef.
Sentinel	49.000
Sinbad	59.000
Skyfox II	59.000
Starfleet	59.000
Starglider II	49.000
Superman	39.000
Sword of sodan	69.000
The bard's tale II	49.000
The works!	249.000
Three Stooges	59.000
Tracker	49.000
TV Sports: Football	59.000
Ultima IV	49.000
Universal military sim.	45.000
Data disks 1 e 2	telef.
Victory road	39.000
Videoscape 3D 2.0	250.000
Designs disks	telef.
VIP professional	199.000
Virus	29.000
Virus infection protection	89.000
WB Extras	69.000
Whirligig	39.000
Willow	59.000
World Cl. Leaderboard	25.000
Zak McKracken	59.000
Zing keys	25.000
Zoetrope	199.000
Zoom	45.000
3 Demon	145.000

COMMODORE 64 CASS.

After burner	18.000
Barbarian II	18.000
Batman	18.000
Dragon ninja	15.000
Exploding fist+	15.000
G.I.hero	15.000
Italy 90 soccer	18.000
Last Ninja II	25.000
MicroSoccer	telef.
Robocop	18.000
R-type	18.000
Serve & volley	22.000
Shoot'em up constr. kit	25.000
Stunt bike simulator	5.000
Tiger road	15.000
Total eclipse (ital.)	15.000
4 soccer simulator	18.000

COMMODORE 64 DISCO

ADD: Heroes of the lance	telef.
ADD: Pool of radiance	59.000
American civil war III	49.000
Barbarian II	25.000
Deathlord	35.000
Driller	29.000
Echelon (con LipStick)	59.000
Eternal dagger	35.000
Fast break	29.000
Fish	29.000
Game over I+II	25.000
GeoPublisher	100.000
Grand Prix Circuit	telef.
Gulf strike	49.000
Home video producer	69.000
Italy 90 soccer	25.000
LED storm	15.000
Mars saga	35.000
McArhtu's war	49.000
MicroSoccer	telef.
Neuromancer	39.000
One on one II	29.000
Powerplay hockey	29.000
Red storm rising	49.000
Rocket Ranger	telef.
Shoot'em up constr. kit	35.000
Stealth mission	75.000
T.K.O.	29.000
The bard's tale III	35.000
Total eclipse (ital.)	20.000
Ultima V	49.000
Wasteland	39.000
Wec Les Mans	21.000
Who framed Roger Rabbit	telef.
Zak Mc Kracken	39.000
4 soccer simulator	25.000

COMMODORE 128 DISCO (128K, 80 COL.)

"C" Language	99.000
Basic 8.0	75.000
GEOS 128	telef.
Mach 128	125.000

OFFERTE SPECIALI

(FINO AD ESAURIMENTO)

FIREBIRD SPECIAL		Valore	A sole
AMIGA	Bubble Bobble+Whirligig+Enlightenment	107.000	75.000
C64 cass.	Savage+Intensity+Sentinel	48.000	33.000
C64 disco	Savage+Intensity+Sentinel	65.000	39.000
AEGIS ENTERTAINMENT			
AMIGA	Arazok's Tomb+Ports of Call	120.000	89.000

DEFENDER OF THE CROWN C64: cass. 12.000 disco 15.000

... SOFTMAIL TI DA' DI PIU'.

BUONO D'ORDINE da inviare a: **LAGO DIVISIONE SOFTMAIL**, VIA NAPOLEONA 16, 22100 COMO, FAX (031) 30.02.14, TEL (031) 30.01.74

Desidero ricevere i seguenti articoli:

☐ Addebitate l'importo sulla mia CARTASI/MASTERCARD/VISA/AMERICAN EXPRESS nr. ______ scadenza ______

☐ Pagherò al postino in contrassegno

Titolo del programma	Computer	Cassetta/disco/accessorio	Prezzo
		Contributo spese di spedizione Lit.	5.000
		TOTALE LIT.	

ORDINE MINIMO LIT. 20.000 (SPESE ESCLUSE) AMJ

Cognome e nome ______

Indirizzo ______ Nr. ______

CAP ______ Città ______ Prov. ______ Telefono ______

FIRMA (SE MINORENNE QUELLA DI UN GENITORE)

VERRANNO EVASI SOLO GLI ORDINI FIRMATI ______

AMJ5

di Luigi Manzo e Giovanni Michelon

Iniziamo da questo numero un corso dedicato all'illustrazione delle caratteristiche del Modula-2, un linguaggio creato in tempi relativamente recenti da N. Wirth ed erede del Pascal.

Si tratta, come noto, di un linguaggio "general purpose", ad alto livello ma con ampie possibilità di controllo delle strutture fondamentali dalla macchina; esso si pone, a nostro parere, come unica alternativa valida all'utilizzo esteso del "C" sia su microcomputer che su grossi sistemi.

Siamo stati indotti a dedicargli ampio spazio su questa rivista anche grazie all'esistenza di una sua ottima implementazione su Amiga, dovuta alla TDI, che possiamo senza dubbio ritenere per molti aspetti superiore ai vari compilatori del mondo MS-DOS da noi provati.

Il corso sarà accessibile a coloro che non hanno alcuna conoscenza di questo linguaggio; supporremo noti soltanto alcuni concetti fondamentali della programmazione, un minimo di Basic, ed una certa dimestichezza con il CLI di Amiga. In questa primissima parte presenteremo il linguaggio nella sua evoluzione storica e nelle sue linee essenziali, daremo un' occhiata alle caratteristiche dell'implementazione della TDI, e vi suggeriamo una possibile struttura per un disco di lavoro con questo compilatore.

Auguriamo a tutti una proficua lettura!

Un po' di storia

Il Modula-2 è un linguaggio per utilizzo generale, nato nel 1979 ad opera del professor Nikolaus Wirth e di un gruppo di

suoi collaboratori dell'università di Zurigo. Esso appartiene a quella categoria di linguaggi che hanno il loro capostipite nell'Algol, nato nel lontano 1959; le strutture fondamentali di questo oramai poco utilizzato linguaggio passarono, dopo le varie riedizioni del '60 e del '65, nel Pascal, nato nel 1970 ad opera del Wirth. In questo, trovano realizzazione pratica alcuni concetti fondamentali presenti in molti linguaggi moderni ad alto livello: tipizzazione estesa delle variabili, rigorosa strutturazione, procedure, eccetera.

In particolare, un erede diretto del Pascal è proprio il nostro Modula-2, la cui prima versione, che ha il nome semplicemente di "Modula", nasce nel 1977.

Il prof. Wirth la elaborò soprattutto con l'intento di eliminare alcune limitazioni più vistose del Pascal standard. In realtà, il Pascal nacque soprattutto per esigenze didattiche, ed è inutilizzabile per lo sviluppo di grossi sistemi: innanzitutto, esso non è pensato come linguaggio modulare, cioè con unità di programma compilabili separatamente, poi non permette la programmazione a basso livello, indispensabile per il controllo completo di un calcolatore. Proprio nel miglioramento di questi due aspetti possiamo cogliere le novità sostanziali del Modula-2 rispetto al suo progenitore.

Per quanto riguarda invece la sostanza delle strutture sintattiche e grammaticali, esse sono state riprese pressochè integralmente dal vecchio linguaggio, ed un listato scritto in Modula-2 risulta quasi interamante comprensibile a chi abbia un minimo di esperienza con il Pascal.

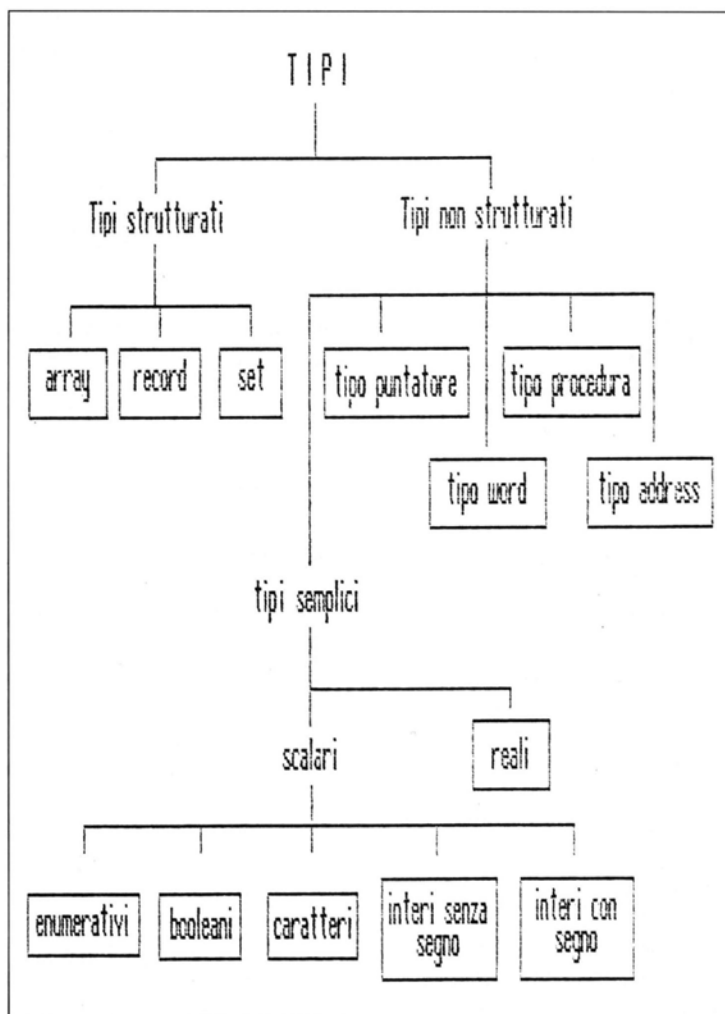
La prima versione del Modula non ebbe praticamente alcuna diffusione, soprattutto perché i primi compilatori furono concepiti per un calcolatore dedicato e risultarono praticamente non trasportabili su altre macchine: soltanto alcuni anni dopo, il linguaggio, ulteriormente rivisto e ribattezzato Modula-2, esce dall'ambito strettamente sperimentale per diventare di dominio pubblico con le implementazioni sui vari micro.

Al 1983 risalgono le pubblicazioni di Wirth relative alle specifiche del linguaggio, alle quali si deve fare riferimento come standard (vedere bibliografia riportata in fondo all'articolo). Attualmente le caratteristiche del linguaggio risultano abbastanza ben definite, anche grazie al fatto che esso tiene conto dell'esperienza fatta con il Pascal; tuttavia esso è ancora molto giovane e non può contare sulla grande messe di pubblicazioni, compilatori, strumenti software e programmi realizzati di cui godono invece altri linguaggi da più tempo affermati. Speriamo, da queste pagine, di poter dare un piccolo contributo alla diffusione di questo linguaggio ineguagliato, a nostro parere, per potenza e chiarezza strutturale.

Filosofia del linguaggio

Lo strumento di programmazione fondamentale offertoci dal Modula-2 è, come ci conferma il suo stesso nome, la possibilità di strutturare i programmi in blocchi, o "moduli", distinti e compilabili separatamente; ciò offre l'immediato vantaggio di poter modificare parte di un programma complesso senza doverlo poi compilare tutto daccapo, ma limitando tale operazione al solo modulo interessato alla modifica.

Tale possibilità non è ovviamente esclusiva del Modula-2; ogni linguaggio pensato per la progettazione di sistemi estesi, a partire dal vecchissimo Fortran al non più giovane C, la offre. Tuttavia, Modula-2 ci consente una serie di facilitazioni che riguardano la possibilità di costruire moduli di libreria, e soprattutto di modificarli in tempi successivi, senza dover preoccuparsi di dover modificare il programma principale che li utilizza; tutto ciò, mantenendo sempre un controllo formale sulla correttezza degli oggetti importati.

Per raggiungere tale obbiettivo vengono utilizzati dei cosiddetti "moduli di in-

Schema riassuntivo dei tipi disponibili in Modula 2

Simbolo	Nome	Significato
+	segno più	addizione ed unione di insiemi
−	segno meno	sottrazione o differenza di insiemi
*	asterisco	moltiplicazione e intersezione di insiemi
/	slash	divisione e differenza simmetrica di insiemi
:=	due punti e uguale	usato per gli assegnamenti
&	e commerciale	sinonimo di AND
=	segno uguale	uguaglianza logica
<>	simboli minore e maggiore	non uguaglianza
#	cancelletto o diesis	sinonimo di #, da preferirsi
<	simbolo minore	minore di
>	simbolo maggiore	maggiore di
<=	minore e uguale	minore o uguale a
>=	maggiore e uguale	maggiore o uguale a
( )	parentesi tonde	usate per racchiudere gli argomenti delle procedure e nella definizione dei tipi enumerativi
[ ]	parentesi quadre	usate per racchiudere gli indici delle tabelle e nella definizione dei subranges
{ }	parentesi graffe	usate per racchiudere gli elementi di un insieme
(* *)	parentesi con asterisco	usate per racchiudere i commenti
^	simbolo di esponente	usato per i puntatori
,	virgola	separa una lista di identificatori similari
.	punto	usato nell'ultimo END di un modulo, ne indica la fine; oppure per indicare un numero reale
;	punto e virgola	indica la fine di una istruzione, o separa i parametri nella definizione di procedura
:	due punti	usato per definire una variabile e per identificare una voce di CASE
|	barra verticale	termina un voce di CASE
..	due punti di seguito	usato nella specificazione dei subranges
'	apostrofo	usato per delimitare una stringa
"	virgolette	usate per delimitare una stringa

Punteggiatura del Modula 2

terfaccia"; in pratica, ad ogni modulo di libreria in cui vengono costruiti effettivamente gli oggetti di cui si ha bisogno (possono essere delle costanti, delle strutture dati, delle procedure...) viene affiancata una di queste interfacce con la funzione di vera e propria "carta di identità" del modulo stesso. In essa infatti sono elencati e descritti per sommi capi gli oggetti contenuti nella libreria senza preoccuparsi del modo in cui sono effettivamente costruiti.

In fase di compilazione del programma principale, il compilatore andrà a consultare proprio tali interfacce per verificare la correttezza di utilizzo degli oggetti usati. Ovviamente, tale carta di identità non potrà essere falsificata; il compilatore, all'atto della sua creazione, andrà a verificare che ciò che viene dichiarato corrisponda effettivamente al contenuto della libreria, segnalando prontamente ogni incongruenza.

Bisogna evidenziare inoltre che le librerie possono essere scritte in un qualunque altro linguaggio, purchè vengano rispettate alcune modalità di scambio di dati tra i vari moduli; ad esempio, nel pacchetto fornito dalla TDI per l'implementazione del linguaggio, di cui parleremo più avanti ed a cui faremo costantemente riferimento, è compresa una serie di moduli di libreria scritti in Modula che a loro volta richiamano le routine scritte in assembly ed in C del sistema operativo e di Intuition.

Un'altra caratteristica ereditata dal Pascal è il rigoroso controllo delle variabili, suddivise in categorie, o "tipi"; variabili di diverso tipo non possono essere mescolate o alterate da funzioni inappropriate. Il Modula-2 comunque, a differenza del Pascal, permette delle assegnazioni cosiddette forzate: tale tecnica, indispensabile in talune applicazioni di basso livello, è tuttavia sconsigliabile a programmatori che non conoscano la struttura della macchina su cui stanno lavorando. In realtà, la "tipizzazione" delle variabili, si rivela uno

strumento prezioso che permette al compilatore e, alla lunga, al programmatore di rivelare errori altrimenti piuttosto insidiosi da scorgere.

Tutto ciò permette di costruire, con notevoli semplificazioni, programmi anche molto lunghi e complessi, anche perché viene facilitata la simbiosi tra diversi programmatori che possono sviluppare indipendentemente i loro moduli.

Ulteriori pregi del Modula-2 sono:

- linguaggio "piccolo", implementabile cioè su piccoli sistemi quali i microcalcolatori.
- regole di sintassi semplici e consistenti; ciò implica una notevole facilità di apprendimento da parte del programmatore ed una maggiore leggibilità dei programmi.
- strutturazione; è un linguaggio procedurale, cioè i vari compiti svolti da un programma possono essere ripartiti in tante piccole unità (procedure appunto) relativamente indipendenti le une dalle altre. Ciò significa che un dato problema può essere concettualmente e praticamente suddiviso in tanti sottoproblemi più semplici da affrontare; inoltre viene semplificata la fase di ricerca degli errori.
- tipizzazione; come abbiamo detto, le variabili sono divise in tipi.

La combinazione di diversi tipi permette la costruzione di strutture dati complesse. In figura 1 è riportato un albero in cui sono riportati tutti i tipi fondamentali disponibili nel linguaggio; li illustreremo uno ad uno, a partire dalla prossima puntata.

Convenzioni ortografiche Modula-2

Come tutti i linguaggi dotati di un certa personalità, anche il Modula-2 è dotato di uno stile tipografico. Ognuno è, ovviamente, libero di scegliere il look dei propri programmi, noi, seguendo il consiglio di più autori, intendiamo comunque descrivere quelle che sono le notazioni più usate nel redigere programmi strutturati e Modula-2. I pregi della standardizzazione sono noti: pensate di aver a che fare con persone diverse, le quali invece del metro usano i pollici o le spanne o il miglio (quello marino, o quello terrestre inglese o americano?).

Convinti che comunque ci sarà sempre qualcuno, che userà le yarde perché più comode, vediamo di descrivere l'ortografia del Modula-2; i caratteri adottati sono lo standard ASCII, codificati da 0 a 127 (i caratteri da 128 a 255 variano da sistema operativo ad un'altro, nonostante l'IBM cerchi di fissare a norma il proprio ASCII esteso).

Il Modula-2, così come è definito dal suo ideatore, utilizza sia i caratteri minuscoli che quelli maiuscoli, cosicchè "Ch" è diverso da "ch" (ecc.), bisogna perciò far attenzione a come è scritta una variabile; non c'è poi limitazione nella lunghezza del nome della variabile, non solo: tutti i caratteri sono ritenuti validi, non solo i primi due, come nel Basic del Commodore 64, o solo i primi otto come nello standard Pascal.

Ecco quindi una prima convenzione: usare per le variabili nomi estesi, che ne ricordino il significato; per esempio al posto di "TN" conviene decisamente usare "TabellaNomi" o "TabNomi"; naturalmente senza farsi prendere dall'entusiasmo: "TabellaDeiNomiCheMiPiaccionoTanto", diventa decisamente scomodo da leggere e da usare.

Una seconda convenzione, già implicita negli esempi appena visti, consiste nello scrivere in maiuscolo la prima lettera di ogni parola che, accostata ad altre, compone il nome di una variabile. Se vi è difficile da ricordare, pensate che Wirth è tedesco, e che in quella lingua tutti i sostantivi si scrivono con iniziale maiuscola.

I testi dei programmi si scrivono con un editor (come Ed); non c'è alcun limite alla lunghezza delle linee del programma, in realtà il Modula-2 considera il testo del programma come se fosse un'unica stringa.

Questo non è un invito a scrivere linee piene zeppe di istruzioni, come spesso si vede in Basic, ma anzi vuole essere la dimostrazione che il listato serve al programmatore e alla sua comodità di comprenderne immediatamente il significato; scrivere perciò male serve solo ad aumentare le possibilità di commettere errori e di non capirci più nulla quando, fra un paio di mesi, vorrete modificare qualcosa.

La punteggiatura è poi praticamente quella del Pascal, che del resto è simile a quella del Basic, ne riportiamo una tabella riassuntiva in figura 2; naturalmente spiegheremo man mano le differenze con gli altri linguaggi. Per il momento basti sapere che i commenti si racchiudono fra "(*" e "*)"; che l'assegnazione delle variabili avviene con ":=", per esempio a:=b*4

```
AND             ELSIF            LOOP          REPEAT
ARRAY           END              MOD           RETURN
BEGIN           EXIT             MODULE        SET
BY              EXPORT           NOT           THEN
CASE            FOR              OF            TO
CONST           FROM             OR            TYPE
DEFINITION      IF               POINTER       UNTIL
DIV             IMPLEMENTATION   PROCEDURE     VAR
DO              IMPORT           QUALIFIED     WHILE
ELSE            IN               RECORD        WITH
EXPORT e QUALIFIED sono mantenute per compatibilita'
```

Parole riservate del Modula 2

4.0) Directory del disco per il Modula-2

```
C (dir)
  contiene i comandi copiati dal disco Workbench,
  il compilatore Modula e il Link
  copiate anche SetMap da System e mettetelo qui
L (dir)
  contiene le librerie prese dal disco Workbench
DEVS (dir)
  keymaps (dir)
      i
  printers (dir)
      generic      (* o quella che preferite *)
  clipboards (dir)
  clipboard.device              jdisk.device
  MountList                     narrator.device
  parallel.device               printer.device
  serial.device                 system-configuration
S (dir)
  StartUp-Sequence
Mods (dir)
  conterrà i files .mod
Defs (dir)
  conterrà i files .def
SymLnk (dir)
  conterrà i files .sym e .lnk
Programs (dir)
  conterrà i programmi eseguibili
```

4.1) Esempio di StartUp-Sequence

```
BindDrivers
SetClock opt load       ;non su A 500
AddBuffers df0: 30
SetMap i
Dir RAM:
Path RAM: add
MakeDir RAM:C
MakeDir RAM:T
Stack 20000
Copy C:Copy to RAM:C
Assign C: RAM:C
CD SYS:C
Copy Dir to C:
Copy List to C:
Copy Ed to C:     ;o TxEd o l'editor che preferite
Copy CD to C:
Copy Delete to C:
Copy Assign to C:
Copy If to C:
Copy Else to C:
Copy EndIf to C:
Copy Link to C:
Copy Modula to C: ; non su A 500
Assign T: RAM:T
Assign M: SYS:Mods
Assign D: SYS:Defs
Assign SL: SYS:SymLnk
Assign P: SYS:Programs
```

ha lo stesso significato del LET a=b*4 (che nessuno più usa) in Basic; che le istruzioni vanno terminate con un punto e virgola.

Ultima cosa, il Modula-2 possiede 40 parole chiave (o parole riservate), che non possono essere ridefinite e che vanno scritte in maiuscolo come riportato in figura 3. Si noti che, a differenza dell'AmigaBasic, si può usare una variabile il cui nome abbia la sventura di contenere una parola riservata; cioè MODella è un identificatore lecito in Modula, non altrettato in AmigaBasic.

Implementazione TDI

L'unico compilatore Modula per Amiga, che si sia visto in Italia, è il prodotto della TDI, (ora Modula-2 Software), che, forse perché senza concorrenti, è decisamente buono.

Beh, al solito stiamo scherzando è vero che la concorrenza spinge a produrre opere migliori, però e altrettanto vero che il pacchetto della TDI merita le nostre lodi, proveremo ciò con dei benchmarks, facendo confronti con i compilatori C.

Quindi da ora in poi, quando parleremo di compilatore Modula per Amiga, ci riferiremo a quello della TDI.

La TDI ha messo sul mercato, ormai da tempo la versione 3.00a del compilatore; poichè, molto probabilmente, circoleranno ancora molte copie della versione 2.00, cercheremo di tenerla presente, pur utilizzando nelle nostre prove la nuova release.

Innanzi tutto va detto che la TDI ha realizzato una piena implementazione del linguaggio, includente diverse caratteristiche come le cooroutine e l'utilizzazione delle interruzioni (interrupt); come già detto più sopra è previsto l'aggancio a Intuition e a tutte le ROM Kernel routines.

La minima richiesta di sistema è 512K e un disk drive, se però disponete di soli 512K, allora è consigliabile avere due drive; con Amiga 2000 si riesce a lavorare benissimo (per le pretese di questo corso) con un solo drive.

Sia la versione 2.00 che 3.00 sono disponibili in due pacchetti diversi:

la cosiddetta versione regolare che comprende due dischetti con compilatore, linker, librerie ed alcuni demo;

la versione "denveloper" con tre dischetti, che contiene inoltre i sorgenti delle interfacce dei moduli di libreria, moduli addizionali per l'utilizzo del IFF e del ILBM, e altre utility come decodificatori dei files di libreria e un programma per il cross reference. Il primo dischetto contiene i files di libreria in una sotto-directory chiamata M2, nella versione 3.00 tali files sono ben 205; vi mostreremo nel corso di questa nostra chiacchierata sul Modula-2 come utilizzarli.

I dischi non contengono il sistema, percui dovrete far partire la vostra Amiga con il WorkBench o con un disco preparato appositamente come vi suggeriremo fra un po'. La versione 3.00 è, come si può leggere nelle note di realizzazione presenti sul dischetto, un vero compilatore a 32 bit, percui si possono dichiarare tabelle da un mega (2^20 = 1024*1024) di elementi, oltre alla possibilità di utilizzare variabili sia intere che reali in doppia precisione. Descriveremo più in dettaglio le caratteristiche proprie di questo prodotto nel prosieguo del corso, in modo da metterne meglio in luce pregi e difetti.

Bachi ed altri insetti

Come in tutti i prodotti di una certa complessità, anche in questo si nascondono dei bugs; per il momento non ne abbiamo notati nella versione 3.00, quelli sinora noti della versione 2.00 sono:

```
VAR n:INTEGER;
....
n:=5;
WriteInt(-5+n,1);
```

ora il risultato dovrebbe essere la scrittura del numero zero, mentre viene scritto "-10", cioè il compilatore traduce "-n+5" come "-(n+5)".

Un altro baco è il seguente:

```
IF Foo(x) > Foo(y) THEN
  WriteString("Foo(x) > Foo(y)");
  WriteLn;
END;
```

Ora anche se Foo(x) restituisce un valore maggiore di Foo(y), il compilatore valuta comunque come falsa l'espressione e il blocco di istruzioni fra THEN e END non viene mai eseguito.

Probabilmente ci saranno altri errori nascosti in altri moduli; però, cosa che si sembra doveroso sottolineare, la struttura del Modula-2 ne permette una facile correzione. Infatti se in un modulo viene trovato un errore è sufficiente correggerlo lasciando inalterata l'interfaccia del modulo stesso; basta cioè far circolare la nuova versione di solo quel modulo, e non come avviene con il C dove, qualora ci sia un baco in un file di libreria, si è costretti ad attendere la nuova versione dell'intera libreria.

Un disco per il Modula-2

In base alla nostra personale esperienza abbiamo trovato utile preparare un dischetto per il Modula simile a quello che ora descriviamo.

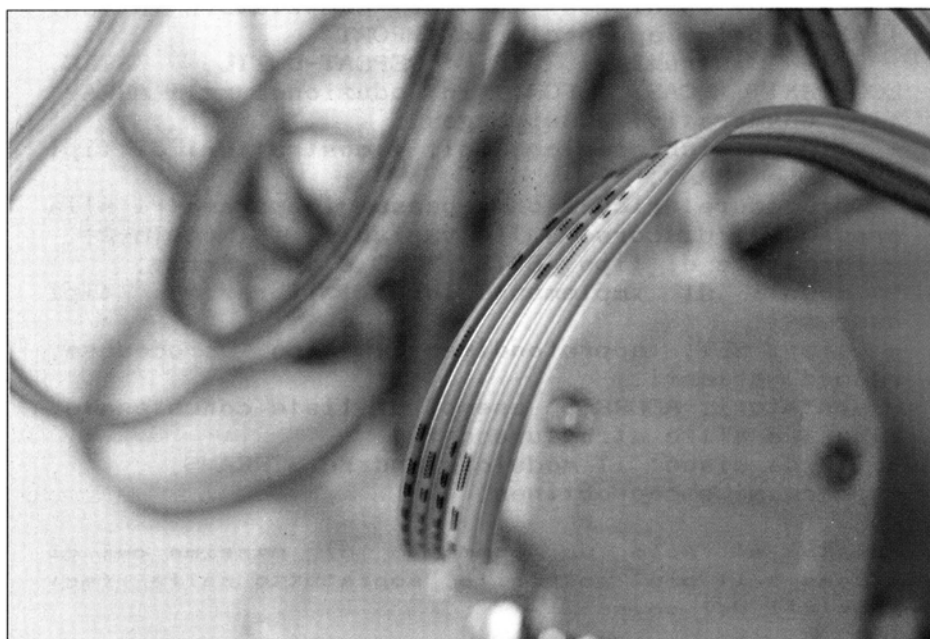
Prima però di iniziare vi raccomandiamo di fare una copia di sicurezza dei dischi Modula-2 (che non sono protetti), tale raccomandazione è quasi un diktat per il disco con le librerie: non rischiate sciocccamente di rovinarlo!

Allora formattate e installate (comando INSTALL) un dischetto, quindi preparatelo come segue (osservate la figura 4.0 dove è riportato un facsimile del contenuto del dischetto):

- copiate le sotto-directory C, L dal disco WorkBench, nella C copiate SetMap (da System) e anche il compilatore (si chiama Modula) e il linker (Link) che si trovano nella directory C di uno dei dischetti TDI;
- preparate le directory S e DEVS come in figura;
- aprite le nuove directory: Mods, Defs, SymLnk, Programs; queste directory conterranno rispettivamente, i sorgenti dei files .mod, i sorgenti dei files .def, i files simbolici .sym delle interfacce e i files .lnk delle librerie, per ultimo i programmi eseguibili. Veniamo ora alla StartUp-Sequence, noi suggeriamo quella di figura 4.1 con evidenziate le differenze fra Amiga 500 e 2000, la cosa importante da far notare è la presenza del comando Stack 20000, che serve al compilatore per funzionare correttamente.

Come potete osservare c'è una lunga serie di ASSIGN; questo vi permetterà di utilizzare le comode abbreviazioni M: D: SL: P: per spostare i vostri files nelle loro giuste sedi.

Struttura del corso

Come abbiamo già detto in apertura, questo corso sul Modula-2 è dedicato ai

neofiti del linguaggio, che, si suppone, abbiano solo qualche conoscienza del Basic e sappiano cavarsela con il CLI; non ce ne vogliano perciò i lettori più smaliziati, se qualche volta sembreremo noiosi e pedanti.

Un altro motivo che ci induce a questa scelta di impostazione è il fatto che il Modula-2, oltre ad essere un linguaggio complesso, è anche decisamente nuovo al grande pubblico degli "smanettoni".

Poichè però ci rendiamo conto, che molti possano trovare stupido dover attendere una decina di lezioni prima di trovare ciò che loro interessa, pubblicheremo, parallelamente al corso, ma al di fuori di esso, dei programmi o moduli di utilità generale, che sfruttino le possibilità e del linguaggio in sè e del suo interfacciamento con Amiga.

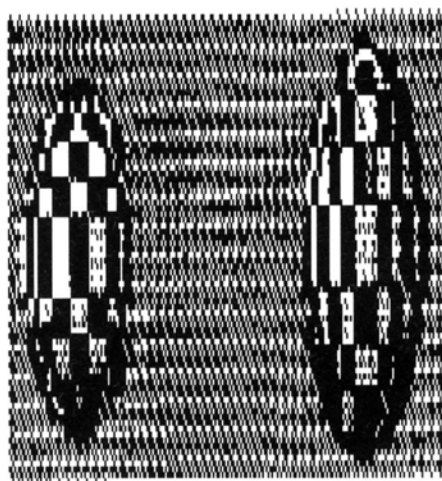
E ovvio che tali programmi torneranno in seguito utili anche a tutti coloro che seguiranno fedelmente il corso, che così non si vedranno costretti ad attendere altre venti lezioni prima di poter effettivamente utilizzare il Modula su Amiga.

Come abbiamo già promesso, esamineremo, all'interno del corso, vari moduli di libreria, in modo da permettere al nostro lettore di impadronirsi di quelle conoscenze necessarie per sviluppare propri programmi sfruttando appieno le possibilità della macchina e del software che l'accompagna.

Per la gioia dei nostri ventiquattro lettori diamo, in tabella 5 un sommario di massima, che cercheremo di seguire nello sviluppo di questa rubrica dedicata a Modula-2; gli argomenti sono raccolti a senso e non secondo le puntate. Questo sommario vuole servire da riferimento per coloro che ci seguiranno, che saranno così facilitati nel comprendere la struttura del Modula, dai suoi elementi base e comuni agli altri linguaggi alle sue caratteristiche più personali.

Come nostra consuetudine, saremo ben felici di rispondere a domande o proposte dei lettori (nel limite delle nostre umane possibilità).

Bibliografia

Il testo fondamentale di riferimento per il linguaggio, di cui esistono due versioni, è

N. WIRTH:

Programming in Modula-2
Springer-Verlag, 1983

Consigliamo vivamente, dello stesso Wirth, il testo (in italiano):

Algoritmi + strutture dati = programmi
Tecniche Nuove, 1987

Esso tratta più in generale dei problemi fondamentali della programmazione; risulta di una chiarezza e di una semplicità auree, e ne esiste una edizione con esempi svolti in Modula-2

Un buon testo, chiaro e ben strutturato, di introduzione e consultazione per questo linguaggio, è

J. B. MOORE K. N. MCKAY:
Modula-2 Text and Reference
Prentice-Hall
disponibile però solo in lingua inglese

Infine faremo spesso riferimento alle "bibbie" di Amiga:

AMIGA Intuition Reference Manual
Commadore Business Machines, Inc.

AMIGA ROM Kernel
Reference Manual: Exec
Commadore Business Machines, Inc.

AMIGA ROM Kernel
Reference Manual: Libraries and Devices

Sommario di massima del corso Modula-2

- descrizione dei cinque tipi base e delle operazioni su di essi; uso del compilatore e del linker; primo approccio ad IMPORT;
- le istruzioni IF, REPEAT-UNTIL, WHILE, LOOP-EXIT, CASE, FOR; introduzione agli ARRAY; tipi enumerativi e subranges;
- tipi, dichiarazione di; i costruttori di tipi RECORD e ARRAY;
- procedure; ricorsione; passaggio parametri alla procedure; discorso sulle variabili in Modula-2;
- forzature di tipo: BYTE, WORD e open array;
- moduli di implementazione; librerie; tipi nascosti;
- tipo SET; approfondimenti: tipo procedura, record varianti;
- puntatori, ADDRESS; esempi di liste concatenate, alberi e altre strutture dati;
- I/O da disco; il modulo standard STREAMS;
- processi e cooroutines;

Questo e' solo un sommario di massima cui ci atteremo il piu' possibile, sopratutto nelle fasi iniziali del corso.

ARRAY IN C

Di Fabio Biancotto

Uno dei vantaggi del disco dell'Amiga contenente i comandi di sistema è la facilità con la quale i comandi DOS esistenti possono essere modificati e addizionati di altri nuovi. Con un semplice esempio, per chi usa l'MS-DOS è possibile usare il comando con il nome "CANCELLARE", per chi lo preferisce, al posto di "DELETE" , cambiando semplicemente il nome del programma delete nella directory comandi con questo comando:

```
RENAME WorkBench:c/delete as WorkBench:c/CANCELLARE
```

Creare un nuovo comando AmigaDOS è veramente semplice, dobbiamo aggiungere un nuovo programma compilato nella directory dei comandi. Questo fa il programma "esecutibile" in qualsiasi momento, appena vogliamo caricare qualche altro comando AmigaDOS, senza occuparci di controllare dov'è la directory comandi, se è la stessa o se è istallata nel disco WorkBench. L'Amiga si prende cura di ogni cosa, comunicandoci di inserire il nostro disco sistema quando è necessario.

Naturalmente, non tutti i programmi aggiunti alla directory comandi possono essere considerati nuovi comandi AmigaDOS. Un esempio è un game arcade chiamato GALAXY_CRUNCH, perché no? Bene, non possiamo considerare un game arcade come un nuovo comando perché no e un utility si o no! Non fa qualcosa associandosi al sistema dei comandi del computer.

Vorremmo un programma calcolatrice tipo NUMERI? Può essere un comando valido? Sicuro! Un calcolatore che operi nella finestra CLI; ci piacerebbe. Potremmo digitare NUMERI 345*894/TAN(43.7)+EXP(28.345) e l'AmigaDOS potrebbe darci ... uh ... -1059184.709 (con l'argomento della TANgente in radianti). Questo potrebbe veramente aiutare uno studente e potrebbe dire aggiungo il comando calcolatrice NUMERI al mio disco sistema.

Per questo articolo, abbiamo scritto un semplice programma chiamato ROMANO che ci da l'equivalente del numero romano da un numero arabico. Credetici o no, ma in certi sistemi operativi questo comando esiste già ed è questo che mi ha dato l'idea di farne una versione per l'Amiga in C. Semplicemente digitando nella finestra CLI:

```
ROMANO 29
```

otteremo dall'Amiga:

```
XXIX
```

Però aspettiamo un minuto. Digitare "ROMANO 29" è più complicato che non digitare semplicemente il nome del programma residente nella directory dei comandi. In qualche modo il programma deve riconoscere il numero 29, o ogni altro numero di quelli che deve cercare nei numeri romani. Questo è chiamato "passaggio di linea comando" ed è questo il vero oggetto di questo articolo.

Ogni volta che un programma è eseguito da una finestra CLI dell'AmigaDOS, questo programma ha qualcosa da scoprire:

1 Quante parole ha digitato l'utente?
2 Quali sono queste parole?

Queste informazioni sono passate nella funzione main() del nostro programma; per prendere questa informazione, dobbiamo usare due variabili. Possiamo in generale scrivere:

```
main(x,y)
```

L'informazione passata potrebbe essere trasmessa sulle variabili x e y. Bene, ma non dimentichiamo che abbiamo propriamente dichiarato queste variabili in conformità a qualche regola del linguaggio C. Per primo, possiamo usare molti nomi di variabili. Possiamo dichiarare la nostra funzione main() del nostro programma come questa:

```
main(quanti,dove_sono)
```

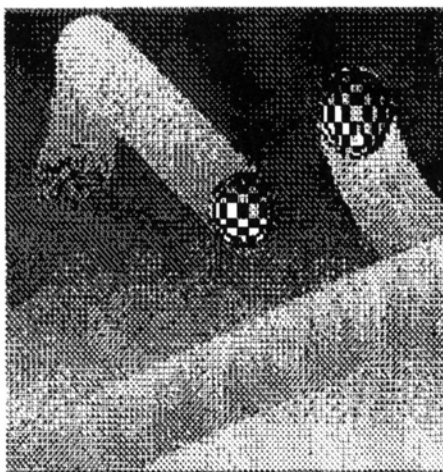
Stiamo considerando il vantaggio del fatto che le implementazioni del Lattice C nell'Amiga ci lasciano nomi di variabili che sono lunghe al massimo 30 caratteri. La prima variabile riceve il numero di parole che sono state passate dal nostro programma. Se noi avevamo digitato questo dalla finestra CLI:

```
ROMANO 35 42 8636
```

Il sistema (AmigaDOS) passerà il numero 4 alla variabile "quanti". Pensiamo che possa essere 3? il sistema ha passato il numero totale di parole che l'utente ha digitato. Il comando ROMANO è considerato una parola. E finalmente, dopo che il sistema ha passato un numero intero, possiamo dichiarare la variabile "quanti" come:

```
main(quanti, dove_sono) nt quanti;
```

Ora il nostro programma può usare la variabile "quanti" per scoprire quanti elementi sono passati nel nostro programma. A che cosa ci può servire questa informazione? Potremmo avere dal nostro programma una risposta di un determinato argomento o la stampa di un messaggio di pericolo se il programma riceve un numero sbagliato di argomenti. Tipo: "Digita solo un numero!" oppure "Ho percepito un sovrabbondante numero di elementi".

Potremmo aver capito che il C è un linguaggio che necessita di puntatori. Puntatori che sono usati per scoprire dove sono le cose. In questo caso, il nostro programma conoscé quanti elementi sono passati, però non conosce ancora esattamente che cosa è passato. Che cosa succede, infatti, alle parole digitate sulla linea del CLI? Sorpresa! Sono convenientemente immagazzinate in memoria per noi dall'AmigaDOS. La locazione in memoria dove questi elementi sono immagazzinate à passata al nostro programma dalla seconda variabile "dove_sono", con un numero che rappresenta il loro indirizzo. Un indirizzo è tipo un numero di locazione di memoria. Diciamo che la variabile "dove_sono" punta la stringa caratteri che l'utente ha digitato nella finestra CLI. Non abbiamo capito esattamente che numero

sia, il nostro programma può leggerlo tutto da solo cercandolo nella variabile "dove_sono".

Così, dobbiamo dichiarare "dove_sono" come un puntatore. Sì, però non è tutto, sappiamo già che l'AmigaDOS permette di ricevere più di un argomento. Questo significha che dobbiamo dichiarare più puntatori? Uno per ogni argomento? Si e no. Il programma ha bisogno di un puntatore separato per ogni argomento passato, però fortunatamente, non dobbiamo dichiararli individualmente. Il sistema che ci aiuterà a finire il tutto è questo: dobbiamo dichiarare un array di puntatori.

Un array è una raccolta di elementi che usano tutti lo stesso nome di variabile, eccetto che ognuna è identificata dal suo o nostro particolare numero, chiamato numero di indice, la dichiarazione:

```
int casa[12];
```

dichiara un array intero di casa[0], casa[1], casa[2],... casa[11].
Questi sono 12 elementi partendo con 0 e finendo con 11. Un array di puntatori e una serie di numeri rappresentanti indirizzi nel computer. Similarmente usano lo stesso nome della variabile seguita da un numero di indice particolare. Dobbiamo dichiarare:

```
char *cerca[20];
```

Questo potrebbe rappresentare 20 posti dove l'indirizzo della stringa caratteri potrebbe essere memorizzata. L'operatore * dice al compilatore che abbiamo solo dichiarato un puntatore. Potremmo memorizzare l'indirizzo del computer del nome della mia migliore amica in "cerca[19]".

```
cerca[19] = &"Alessandra";
```

L'& è l'indirizzo dell'operatore applicato alla stringa "Alessandra". Non sappiamo dove è memorizzato dal compilatore nel computer il nome Alessandra, useremo l'indirizzo dell'operatore per scoprirlo. Potremo ora scrivere:

```
printf("Alessandra è nel mio computer alla locazione
di memoria %u",
cerca[19]);
```

La locazione di Alessandra in memoria sarebbe riportata nella finestra CLI! Wow! Alessandra non sarebbe molto impressionata da tutto ciò ma noi penso che avremmo imparato qualcosa. Finalmente potremmo scrivere:

```
printf("La mia migliore amica è %s",cerca[19]),
```

La finestra CLI risponderà ricordandomi esattamente chi è la mia migliore amica. Alessandra di questo ne sarà felice.

Il comando printf usa l'indirizzo memorizzato nel cerca[19] per cercare e stampare sulo schermo il nome di Alessandra.

Va bene! Va bene! Ritorniamo a parlare di argomenti riguardanti il C.

Dobbiamo dichiarare la variabile "dove_sono" come un array di puntatori. Qui si risolve il tutto con:

```
main(quanti, dove_sono)
int quanti;
char dove_sono[];
```

Non abbiamo dichiarato la grandezza dell'array? No, l'array è passato al nostro programma attraverso l'indirizzo memorizzato. Questo percui sarà l'indirizzo del puntatore della prima variabile "dove_sono[0]". Possiamo scrivere:

```
printf("La prima parola è: %s", dove_sono[0]);
```

La parola romano potrebbe apparire sullo schermo. Perché? perché il primo elemento dell'array puntatore "dove_sono[0]" punta la prima parola digitata dall'utente nella finestra CLI. Questo è il nome del comando, Romano. Il secondo elemento dell'array puntatore "dove_sono[1]" punta la seconda parola digitata dall'utente. Questo accade al primo argomento, un numero in formato carattere, potrebbe essere 29 o 83 o altri. Il sistema separa ogni parola cercando gli spazi vuoti. Qualunque cosa sia separata da spazi vuoti è trattata, e riportata al programma, come una stringa caratteri indirizzata nell'array puntatore "dove_sono[]". Diciamo una stringa caratteri, se abbiamo provato a passare un numero tipo 29, il sistema potrebbe passarci la stringa caratteri "29". Se vogliamo usare questo numero in qualche calcolo, dobbiamo prima averlo convertito nell'intero 29. Questo è fatto nel programma "ROMANO", il quale è nel Listato2 di questo articolo.

Ora conosciamo come scoprire quante parole abbiamo digitato, e dove cercarle nella memoria così potremmo fare qualcosa di affascinante con loro. Possiamo passare qualche argomento al giusto programma nella stessa linea che usiamo per chiamare il programma. Istalliamo il programma nella dirctory comandi WorkBench:c.

Consideriamo il Listato1. Il comando printf ci darà il numero di parole digitate dall'utente sulo schermo. Inoltre, notiamo come il valore trovato nella variabile "quanti" è usato nella parte del loop while, dice di stampare molte parole come sono riportate dal sistema digitato dall'utente. L'indice della variabile i è incrementato da 0 al valore memorizzato in "quanti" e allora causa la fine del loop. Questo è quello che volevamo. Lanciando il programma dalla finestra CLI e digitando qualcosa del tipo:

```
Nome_del_file> Ciao come va la vita Alessandra?
```

e premendo return, otteremo:

```
L'utente ha digitato 7 parole

La parola #1 è <Nome_del_file>

La parola #2 è Ciao

La parola #3 è come

La parola #4 è va

La parola #5 è la

La parola #6 è vita

La parola #7 è Alessandra?
```

Questo è tutto.

Ora prendiamo in esame il programma Romano.c (Listato 2). Questo programma usa esattamente lo stesso metodo per determinare quanti elementi vengono digitati e anche per considerarli uno a uno. Naturalmente, è un po' più complicato di una stampa degli elementi sullo schermo. Inoltre ha un comando che limita il numero massimo di elementi che l'utente può digitare.

Proviamo a domandarci che cosa abbiamo imparato fino ad ora che serva alle nostre applicazioni. Come fare un programma che cerchi la media artimetica di una serie di numeri? O forse un programma che ci dia la dimensione di una fila contando i caratteri che ci sono all'interno? Il comando protrebbe essere chiamato Quanto_Grande e il <Nome_del_file> il file che vogliamo confrontare potrebbe essere l'argomento. Come fare un programma che converta i numeri romani riformandoli in arabici?

Per ora possiamo soltanto guardare il Listato2 per cercare di capire come funziona il programma.

CALENDARIO CORSI 1989

AREA INFORMATICA, AUTOMAZIONE D'UFFICIO E INTELLIGENZA ARTIFICIALE

AUTOMAZIONE D'UFFICIO E AMBIENTE P.C.

CODICE	CORSO	ORE	PREZZO (1)	GENNAIO	FEBBRAIO	MARZO	APRILE	MAGGIO	GIUGNO	LUGLIO	AGOSTO	SETTEMBRE	OTTOBRE	NOVEMBRE	DICEMBRE
N AU-20	Conoscere l'informatica e il P.C.	40	900.000	23-27								4-8			
AU-02	Word	24	450.000			1-3								6-8	
AU-04	Lotus 1-2-3	24	500.000		1-3								2-4		
AU-05	Symphony	40	700.000			6-10								13-17	
AU-06	DBIII Plus utenti	24	450.000	30---------	1								25-27		
N AU-16	Excel	24	700.000						19-21						11-13
N AU-11	Programmazione windows base	80	2.000.000				10-21								11-22

LINGUAGGI DI PROGRAMMAZIONE

CODICE	CORSO	ORE	PREZZO (1)	GENNAIO	FEBBRAIO	MARZO	APRILE	MAGGIO	GIUGNO	LUGLIO	AGOSTO	SETTEMBRE	OTTOBRE	NOVEMBRE	DICEMBRE
PE-02	Basic	40	600.000		20-24								23-27		
PE-03	Pascal-Turbopascal	48	750.000				3-10								11-18
PE-05	Cobol	56	850.000			13-22								20-28	
PE-06	Linguaggio "C"	80	1.400.000	16-27					12-23			11-22			
N PE-06A	Ottimizzazione e debugging "C"	40	1.300.000			6-10								6-10	
PE-08	Prolog-Turboprolog	40	1.200.000							10-14		25-29			
PE-10	Lisp	40	1.050.000					15-19					9-13		

SISTEMI OPERATIVI

CODICE	CORSO	ORE	PREZZO (1)	GENNAIO	FEBBRAIO	MARZO	APRILE	MAGGIO	GIUGNO	LUGLIO	AGOSTO	SETTEMBRE	OTTOBRE	NOVEMBRE	DICEMBRE
AU-01	MS-DOS e ambiente P.C.	24	450.000	13-15					26-28			25-27			
N AU-12	OS/2 - Architettura	40	1.200.000				17-21						23-27		
N AU-25	OS/2 - Presentation manager	40	1.200.000					8-12							18-22
I-09	Unix-Xenix utenti	56	1.400.000					22-30						6-14	

DATA BASE

CODICE	CORSO	ORE	PREZZO (1)	GENNAIO	FEBBRAIO	MARZO	APRILE	MAGGIO	GIUGNO	LUGLIO	AGOSTO	SETTEMBRE	OTTOBRE	NOVEMBRE	DICEMBRE
AU-07	DBIII Plus programmazione	24	500.000				3-5								4-6
N AU-08	DBase IV	32	700.000						19-22			4-7			
N AU-21	Focus utenti	24	700.000					8-10					9-11		
N AU-22	Focus programmazione	40	900.000							3-7				13-17	
N AU-23	Oracle	40	1.000.000						5-7						11-15

METODOLOGIE INFORMATICHE

CODICE	CORSO	ORE	PREZZO (1)	GENNAIO	FEBBRAIO	MARZO	APRILE	MAGGIO	GIUGNO	LUGLIO	AGOSTO	SETTEMBRE	OTTOBRE	NOVEMBRE	DICEMBRE
PE-01	Programmazione elettr. corso base	96	1.500.000			28---	---12					4-19			
I-102	Case	40	2.000.000					8-12					2-6		
I-105	Ingegneria del software	80	2.000.000		6-17							11-23			
N I-I11	Software quality assurance	32	1.200.000		13-16								16-19		
I-108	Architettura SNA	32	1.500.000							5-8				27-30	

INTELLIGENZA ARTIFICIALE

CODICE	CORSO	ORE	PREZZO (1)	GENNAIO	FEBBRAIO	MARZO	APRILE	MAGGIO	GIUGNO	LUGLIO	AGOSTO	SETTEMBRE	OTTOBRE	NOVEMBRE	DICEMBRE
IA-01	Intellig. Artif. - corso base	40	1.400.000				17-21							20-24	

DESK TOP PUBLISHING

CODICE	CORSO	ORE	PREZZO (1)	GENNAIO	FEBBRAIO	MARZO	APRILE	MAGGIO	GIUGNO	LUGLIO	AGOSTO	SETTEMBRE	OTTOBRE	NOVEMBRE	DICEMBRE
DTP-1	Desk top publishing base	40	1.400.000			6-17								27-----	-----1
DTP-2	Ventura	24	600.000						26-28						4-6
DTP-3	Page maker	24	600.000				12-14								18-20
DTP-4	Manuscript	24	650.000					29-31					23-25		

COMPUTER GRAFICA E MEMORIE OTTICHE

CODICE	CORSO	ORE	PREZZO (1)	GENNAIO	FEBBRAIO	MARZO	APRILE	MAGGIO	GIUGNO	LUGLIO	AGOSTO	SETTEMBRE	OTTOBRE	NOVEMBRE	DICEMBRE
N CG-01	Introduz. ai CD-Rom e videodischi	40	1.300.000						5-9			25-29			
N CG-02	Concorde	40	1.000.000		6-10								9-14		
N CG-03	Computer grafica	80	2.000.000				3-14								11-22
AU-10	Autocad	32	900.000			13-16								13-16	

AREA TELECOMUNICAZIONI - TELEMATICA

CODICE	CORSO	ORE	PREZZO (1)	GENNAIO	FEBBRAIO	MARZO	APRILE	MAGGIO	GIUGNO	LUGLIO	AGOSTO	SETTEMBRE	OTTOBRE	NOVEMBRE	DICEMBRE
N T-I1	Il Pabx: Strutture e nuove utilizzazioni	40	1.500.000											6-10	
T-I2	Tecniche base di trasmissione PCM	40	1.500.000												11-15
T-I3	Tecniche base e sistemi per trasmissione dati	80	1.700.000		20------	-3						18-29			
T-I4	Apparati e sistemi per le reti di computer	40	1.500.000			13-17							9-13		
T-I5	Reti a commutazione di pacchetto	40	1.700.000		6-10								16-20		
T-I6	L'integrazione nelle reti di TLC (ISDN, BX)	32	1.800.000											27-30	
T-I7	Servizi a valore aggiunto sulle reti X25	24	1.000.000		15-17										4-6
T-I8	Il modello OSI	32	1.500.000		15-17									20-23	
T-I9	Architettura SNA: principi e applicazioni	40	1.800.000				10-14								

AREA ELETTRONICA

CODICE	CORSO	ORE	PREZZO (1)	GENNAIO	FEBBRAIO	MARZO	APRILE	MAGGIO	GIUGNO	LUGLIO	AGOSTO	SETTEMBRE	OTTOBRE	NOVEMBRE	DICEMBRE
EM 1	Criteri di progettazione analogica	40	700.000	16-20			3-14*		12-16				9-20*		
EM 2	Criteri di progettazione digitale	40	700.000	23-27			17-28*		19-23						
EM 3	Progettazione con micro a 8 bit	40	700.000	30-----3		27---10*			26-30			4-19*			
EM 4	Progettazione con micro a 16 bit	40	1.000.000		6-10		3-14*			3-7				6-17*	
N EM 5	Progettazione con micro a 32 bit	40	1.300.000				3-7			3-14*					
N EM 6	Progettazione con micro single chip	40	1.000.000				17-21								
N EM 7	Periferiche per microprocessori	40	1.000.000				10-14								
N EM 8	Uso e applicazioni delle memorie	40	1.000.000							3-14*					
N EM 9	Collaudo ATE	40	700.000					22-26							
N EM 10	Convertitori A/D D/A	40	700.000						5-16*			4-8			
N EM 11	Progettazione e normative di sicurezza	24	600.000										2-4		
N EM 12	Progettare in alta frequenza	40	700.000			6-10								6-10	
N EM 13	Progettazione alimentatori	20	500.000		6-10*					10-14				6-17*	
N EM 14	Strumentazione e controlli industriali	40	1.000.000							17-21					
N EM 15	Uso di PC industriali	40	1.000.000							24-28				23----------6*	
N EM 16	Video processor	40	1.000.000										2-6		
N EM 17	Progettazione di ciruiti ASIC	40	2.200.000			13-17		8-12	5-16*	10-14		11-15	9-20*	13-17	4-15*
N EM 18	Applicazione delle logiche programmabili	40	1.000.000				8-19*	5-19						13-24*	
N EM 19	I micro e il linguaggio C	40	1.000.000		13-17							11-15			
N EM 20	Processori di segnali digitali	60	2.000.000			14-22						18-27			
N EM 21	Processori a filtri programmabili	40	1.700.000												
N EM 22	Cad Cam	40	1.300.000										16-20		
N EM 23	Reti di microcomputer	40	1.000.000					22-26*							
N EM 24	Progettazione dei BUS	20	700.000			13-17*									
N EM 25	Sistemi operativi per microprocessori	40	1.000.000											20-24*	
N EM 26	Manutenzione Personal Computer	40	1.000.000							3-14*				27-----------1	

AREA AUTOMAZIONE INDUSTRIALE E ROBOTICA

CODICE	CORSO	ORE	PREZZO (1)	GENNAIO	FEBBRAIO	MARZO	APRILE	MAGGIO	GIUGNO	LUGLIO	AGOSTO	SETTEMBRE	OTTOBRE	NOVEMBRE	DICEMBRE
AI&R 1	Specializzazione in automazione e robotica		2.500.000									4---------------------27*			
AI&R 2	Microprocessori a 8 bit	40	700.000									4-19*			
AI&R 3	Traduttori sensori attuatori	20	500.000									21-28*			
AI&R 4	Controllori logici programmabili	40	700.000										2-10*		
AI&R 5	Controllo numerico	20	500.000										19-26*		
AI&R 6	Elementi base di robotica	20	500.000										30----------7*		
AI&R 7	Strutture informatiche nel processo produttivo	20	500.000											9-16*	
AI&R 8	Reti di comunicazione nella fabbrica	20	500.000											20-21*	

AREA ALTE TECNOLOGIE SPECIALI

CODICE	CORSO	ORE	PREZZO (1)	GENNAIO	FEBBRAIO	MARZO	APRILE	MAGGIO	GIUGNO	LUGLIO	AGOSTO	SETTEMBRE	OTTOBRE	NOVEMBRE	DICEMBRE
N ATS 1	Criteri di prova vita sui semiconduttori	40	1.000.000					15-19				11-22*			
N ATS 2	Costi della qualità	8	250.000		22					5					
ATS 3	Tecnologie in fibra ottica per TX dati e immagini	40	1.000.000					8-19*							
ATS 4	Progettazione dei moderni circuiti stampati	40	1.000.000				3-7*								
ATS 5	Affidabilità, circuiti e componenti elettronici	40	1.000.000	13-17*											
ATS 6	Tecnologie VLSI	40	1.300.000												11-15

N = Corsi attivati nell'89.
(1) = Iva esclusa - È compresa la fornitura di testi Jackson e dispense Jackson Sata e servizio mensa.
* = Sono previste sessioni serali in data da definire con gli interessati.

SCUOLA DI ALTE TECNOLOGIE APPLICATE

GRUPPO EDITORIALE JACKSON

S.A.T.A.

HOBBY & HOME COMPUTER

FARE ELETTRONICA *SEMPRE DI PIÚ*

FARE ELETTRONICA prosegue nella serie dei circuiti targati "Elettronica Facile" dei quali viene fornita la basetta stampata in vetronite.
E' la volta di un interessante GONG ELETTRONICO da utilizzare come campanello di casa o come avvistatore acustico general purpose.
Prosegue anche l'inserto "PROGETTARE CON I CIRCUITI INTEGRATI" da conservare e da rilegare con gli altri sei per ottenere il volume completo. Sempre interessanti ed utili, il foglio di acetato con i master delle basette e lo schema TV con le relative sei cartelle di riparazione.

SUPERCOMMODORE C64/128

Oltre all'eccezionale supergame "Mandroid", che vi calerà nei panni di un intrepido 007 sulle tracce di un importante documento sottratto da un collega disertore, SUPERCOMMODORE C64/128 offre un coloratissimo poster da parete, recensioni dei più recenti games, e numerosi programmi registrati su supporto magnetico.

AMIGA MAGAZINE GAMES

Prende il via la rivista dei giochi per Amiga. Subito un game di risonanza mondiale, splendido nella grafica e nell'azione. Gli spacciatori di droga la fanno da padroni: in "Manhattan dealers" li dovrete scoprire ed eliminare.

PUBLISHER: Filippo Canavese

PC GAMES

Sulla rivista di giochi per PC, sono sempre presenti oltre al supergame, tre giochi supplementari, news e coloratissime recensioni di giochi. Con Bob Morane, potrete sfidare le insidie della jungla ed affrontare incredibili avventure.

CEDOLE SERVIZIO LETTORI

ABBONAMENTO RIVISTE JACKSON

ATTENZIONE Questa cartolina riporta sul retro un modulo speciale con una serie di domande a cui preghiamo vivamente di rispondere con precisione.

ORDINI QUI SOTTO CIÒ CHE HA SCELTO

AVVERTENZA I rinnovi entreranno in vigore automaticamente a partire dal numero successivo alla data di scadenza dell'abbonamento precedente.
Per motivi tecnici, i nuovi abbonamenti saranno attivati dopo 6 settimane dalla data di ricevimento della cartolina di richiesta, o in data successiva nel caso di esplicita comunicazione da parte dell'abbonato.

Lo sconto del 40% è stato calcolato, in certi casi, arrotondando le cifre in modo da differenziare le tariffe di ciascuna rivista per esigenze di gestione.
N.B. Per abbonamenti all'estero le tariffe dovranno essere raddoppiate. Non è prevista la spedizione via aerea

☐ NUOVO ABBONAMENTO ☐ RINNOVO

Rivista	Numeri	Prezzo
☐ EO NEWS SETTIMANALE	numeri 40 + 6 omaggio	L. 59.500
☐ ELETTRONICA OGGI	numeri 20	L. 60.500
☐ AUTOMAZIONE OGGI	numeri 20	L. 60.000
☐ MECCANICA OGGI	numeri 40 + 6 omaggio	L. 59.000
☐ STRUMENTAZIONE E MISURE OGGI	numeri 11	L. 39.000
☐ INFORMATICA OGGI SETTIMANALE	numeri 40 + 6 omaggio	L. 61.000
☐ INFORMATICA OGGI MESE	numeri 11	L. 33.500
☐ BIT (quindic. da Gennaio)	numeri 20	L. 48.000
☐ PC MAGAZINE	numeri 11	L. 32.500
☐ PC FLOPPY	numeri 11	L. 79.500
☐ COMPUTERGRAFICA E APPLICAZIONI	numeri 11	L. 39.500
☐ TRASMISSIONE DATI E TELECOM.	numeri 11	L. 34.000
☐ COMPUSCUOLA	numeri 10	L. 24.500
☐ WATT (quindic. da Gennaio)	numeri 20	L. 36.500
☐ MEDIA PRODUCTION	numeri 11	L. 46.500
☐ STRUMENTI MUSICALI	numeri 11	L. 32.000
☐ FARE ELETTRONICA	numeri 12	L. 36.000
☐ AMIGA MAGAZINE disk	numeri 11	L. 92.500
☐ AMIGA TRANSACTOR	numeri 6	L. 25.500
☐ COMMODORE PROFESS. 64/128 disk	numeri 11	L. 85.000
☐ COMMODORE PROFESS. 64/128 cass.	numeri 11	L. 59.500
☐ SUPERCOMMODORE 64/128 disk	numeri 11	L. 79.000
☐ SUPERCOMMODORE 64/128 cassetta	numeri 11	L. 49.500
☐ OLIVETTI PRODEST USER	numeri 6	L. 18.000
☐ PC SOFTWARE	numeri 11	L. 66.000
☐ PC GAMES 5¼"	numeri 11	L. 93.000
☐ PC GAMES 3½"	numeri 11	L. 99.500
☐ 3 ½" SOFTWARE	numeri 11	L. 99.000

MODALITÀ DI PAGAMENTO

☐ Allego assegno n. ______________ di L. ______________
Banca ______________
☐ Ho effettuato versamento di L. ______________ sul c/c postale n. 11666203 intestato a Gruppo Editoriale Jackson - Milano e allego fotocopia della ricevuta.
☐ Ho effettuato versamento di L. ______________ tramite vaglia postale o telegrafico e allego fotocopia della ricevuta.
☐ Vi autorizzo ad addebitare l'importo di L. ______________ sulla carta di credito ☐ VISA ☐ AMERICAN EXPRESS ☐ DINERS CLUB ☐ CARTA SI
N. ______________ Data di scadenza a ______________
Data ______________ Firma ______________

COMMISSIONE LIBRI E GRANDI OPERE

ATTENZIONE Questa cartolina riporta sul retro un modulo speciale con una serie di domande a cui preghiamo vivamente di rispondere con precisione.

ORDINI QUI SOTTO CIÒ CHE HA SCELTO

Codice articolo	Titolo abbreviato	Prezzo L.
	Contributo fisso per spese di spedizione	L. 4.000
	TOTALE	

Ordine minimo L. 40.000 + 4.000 per contributo fisso spese di spedizione
☐ **Non sono abbonato** a riviste Jackson
☐ **Sono abbonato** alla/e seguente/i rivista/e Jackson ______________ e ho quindi diritto allo sconto del 20% sino al 28/2/89 e del 10% dopo tale data

MODALITÀ DI PAGAMENTO

☐ Contro assegno postale ☐ Assegno allegato n° ______________
Banca ______________
☐ Versamento su c/c postale 11666203 a Voi intestato del quale allego fotocopia della ricevuta
☐ Versamento con vaglia postale a Voi intestato del quale allego fotocopia della ricevuta
☐ Carta di Credito ☐ American Express ☐ Visa ☐ Diners Club
☐ Carta Sì Autorizzo l'organizzazione sopra indicata ad addebitare l'importo sul mio Conto n° ______________ Scadenza ______________
☐ Richiedo l'emissione della fattura (formula riservata alle aziende) e comunico il numero di Partita I.V.A. ______________
Data ______________ Firma ______________

ARRETRATI RIVISTE E FASCICOLI GRANDI OPERE

ORDINI QUI SOTTO CIÒ CHE HA SCELTO

ARRETRATI RIVISTE

Rivista	Numero	Anno	Prezzo
☐ INDUSTRIA OGGI	Numero ____	Anno ____	L. 10.000 cad.
☐ ELETTRONICA OGGI	Numero ____	Anno ____	L. 10.000 cad.
☐ AUTOMAZIONE OGGI	Numero ____	Anno ____	L. 10.000 cad.
☐ EO NEWS SETTIMANALE	Numero ____	Anno ____	L. 4.000 cad.
☐ WATT	Numero ____	Anno ____	L. 4.000 cad.
☐ TRASMISSIONE DATI E TELECOMUNICAZIONI	Numero ____	Anno ____	L. 10.000 cad.
☐ VIDEOTEL MAGAZINE	Numero ____	Anno ____	L. 8.000 cad.
☐ INFORMATICA OGGI	Numero ____	Anno ____	L. 10.000 cad.
☐ INFORMATICA OGGI SETTIMANALE	Numero ____	Anno ____	L. 4.000 cad.
☐ LAB NEWS	Numero ____	Anno ____	L. 10.000 cad.
☐ COMPUTER GRAFICA E APPLICAZIONI	Numero ____	Anno ____	L. 12.000 cad.
☐ PC MAGAZINE	Numero ____	Anno ____	L. 10.000 cad.
☐ PC MAGAZINE + PC FLOPPY	Numero ____	Anno ____	L. 24.000 cad.
☐ PC GAMES 5 ¼	Numero ____	Anno ____	L. 24.000 cad.
☐ PC GAMES 3½	Numero ____	Anno ____	L. 30.000 cad.
☐ PC 3½ SOFTWARE	Numero ____	Anno ____	L. 30.000 cad.
☐ BIT	Numero ____	Anno ____	L. 10.000 cad.
☐ SUPERCOMMODORE 64/128 (cassetta)	Numero ____	Anno ____	L. 15.000 cad.
☐ SUPERCOMMODORE 64/128 (disco)	Numero ____	Anno ____	L. 24.000 cad.
☐ NOI 128/64 (cassetta)	Numero ____	Anno ____	L. 18.000 cad.
☐ NOI 128/64 (disco)	Numero ____	Anno ____	L. 26.000 cad.
☐ COMMODORE PROFESSIONAL (cassetta)	Numero ____	Anno ____	L. 18.000 cad.
☐ COMMODORE PROFESSIONAL (disco)	Numero ____	Anno ____	L. 26.000 cad.
☐ AMIGA MAGAZINE (disco)	Numero ____	Anno ____	L. 28.000 cad.
☐ AMIGA TRANSACTOR	Numero ____	Anno ____	L. 14.000 cad.
☐ OLIVETTI PRODEST USER	Numero ____	Anno ____	L. 10.000 cad.
☐ LA RIVISTA DI ATARI	Numero ____	Anno ____	L. 10.000 cad.
☐ COMPUSCUOLA	Numero ____	Anno ____	L. 8.000 cad.
☐ FARE ELETTRONICA	Numero ____	Anno ____	L. 10.000 cad.
☐ LABORATORIO DI ELETTRONICA PROFESSIONALE	Numero ____	Anno ____	L. 3.000 cad.
☐ STRUMENTI MUSICALI	Numero ____	Anno ____	L. 10.000 cad.
☐ NAUTICAL QUARTERLY	Numero ____	Anno ____	L. 20.000 cad.
☐ AUTOMOBILE QUARTERLY	Numero ____	Anno ____	L. 20.000 cad.

ARRETRATI GRANDI OPERE

Opera	Fascicoli	Prezzo	Copertine/raccoglitori	Prezzo
☐ ABC PERSONAL COMPUTER (24 fascicoli 4 copertine)	fasc. n. ____	L. 3.000 cad.	cop. n. ____	L. 7.000 cad.
☐ ABC PERSONAL COMPUTER QUIZ (10 fascicoli 1 copertina)	fasc. n. ____	L. 3.000 cad.	cop. n. ____	L. 7.000 cad.
☐ A SCUOLA DI SCACCHI C64/C128 (10 lezioni 1 raccoglitore)	lez. n. ____	L. 8.000 cad.	racc. n. ____	L. 7.000 cad.
☐ BYTES (60 fascicoli 7 copertine)	fasc. n. ____	L. 3.000 cad.	cop. n. ____	L. 7.000 cad.
☐ CORSO DI GRAFICA C64/C128 (10 lezioni 1 raccoglitore)	lez. n. ____	L. 8.000 cad.	racc. n. ____	L. 7.000 cad.
☐ D.E.I. (110 fascicoli 12 copertine)	fasc. n. ____	L. 4.500 cad.	cop. n. ____	L. 7.000 cad.
☐ ELETTRICITÀ E ENERGIA (52 fascicoli 4 copertine)	fasc. n. ____	L. 3.000 cad.	cop. n. ____	L. 7.000 cad.
☐ E.I. 1a Edizione (51 fascicoli 8 copertine)	fasc. n. ____	L. 3.000 cad.	cop. n. ____	L. 7.000 cad.
☐ E.I. 2a Edizione (61 fascicoli 7 copertine)	fasc. n. ____	L. 3.000 cad.	cop. n. ____	L. 7.000 cad.
☐ E.I. AGGIORNAMENTI (20 fascicoli 2 copertine)	fasc. n. ____	L. 3.000 cad.	cop. n. ____	L. 7.000 cad.
☐ LABORATORIO DI ELETTRONICA (52 fascicoli 5 copertine)	fasc. n. ____	L. 3.000 cad.	cop. n. ____	L. 7.000 cad.
☐ LA GRANDE GUIDA DEL PROGRAMMATORE ☐ 128 ☐ 64 (8 fascicoli)	fasc. n. ____	L. 8.000 cad.		
☐ PC MASTER MS DOS (5 1/4) (9 lezioni 1 copertina)	lez. n. ____	L. 14.000 cad.	cop. n. ____	L. 8.000 cad.
☐ PC MASTER MS DOS (3½) (9 lezioni 1 copertina)	fasc. n. ____	L. 17.000 cad.	cop. n. ____	L. 8.000 cad.
☐ D. BASE III (5 1/4") (8 fascicoli 1 copertina)	fasc. n. ____	L. 14.000 cad.	cop. n. ____	L. 8.000 cad.
☐ D. BASE III (3½) (8 fascicoli 1 copertina)	fasc. n. ____	L. 17.000 cad.	cop. n. ____	L. 8.000 cad.
☐ QUADERNI JACKSON (30 fascicoli)	fasc. n. ____	L. 7.000 cad.		
☐ SOFTWARE (52 fascicoli 5 copertine)	fasc. n. ____	L. 2.500 cad.	cop. n. ____	L. 8.000 cad.
☐ VIDEO BASIC (20 lezioni 5 raccoglitori) ☐ C64 ☐ C128 ☐ VIC20 ☐ MSX	lez. n. ____	L. 8.000 cad.	racc. n. ____	L. 8.000 cad.
☐ 7 NOTE BIT (15 lezioni 3 raccoglitori)	lez. n. ____	L. 10.000 cad.	racc. n. ____	L. 10.000 cad.

MODALITÀ DI PAGAMENTO

NB: Non si effettuano spedizioni contro assegno il pagamento deve sempre essere allegato all'ordine
☐ Allego assegno n° ______________ di L. ______________
Banca ______________
☐ Allego l'importo in contanti di L. ______________
☐ Versamento su c/c postale 11666203 a Voi intestato del quale allego fotocopia della ricevuta

COMPILARE E SPEDIRE IN BUSTA CHIUSA A: GRUPPO EDITORIALE JACKSON - VIA ROSELLINI 12 - 20124 MILANO

SERVIZIO QUALIFICAZIONE LETTORI

ATTENZIONE Questa cartolina riporta un modulo speciale con una serie di domande a cui preghiamo vivamente di rispondere con precisione.

INDIRIZZO PRIVATO

COGNOME E NOME ______

VIA E NUMERO ______

CAP ______ CITTÀ ______ PROV. ______

TEL. (______) ______ ANNO DI NASCITA 19 ______

TITOLO DI STUDIO: ☐ LAUREA ☐ MEDIA SUPERIORE ☐ MEDIA INFERIORE

INDIRIZZO LAVORO

DITTA O ENTE ______

VIA E NUMERO ______

CAP ______ CITTÀ ______ PROV. ______

TEL. (______) ______ TELEX ______

ATTIVITÀ AZIENDA
- **A** ☐ Informatica
- **B** ☐ Automazione Industriale
- **C** ☐ Meccanica
- **D** ☐ Strumentazione elettronica
- **E** ☐ Telecomunicazioni e Telefonia
- **F** ☐ Elettronica
- **G** ☐ Chimica
- **H** ☐ Elettrotecnica e Impianti elettrici
- **I** ☐ Laboratori di analisi Chimica e medica
- **L** ☐ Altra industria manifatturiera
- **M** ☐ Agricoltura
- **N** ☐ Ingegneria/Edilizia/Architettura
- **O** ☐ Finanza/Banche/Assicurazioni
- **P** ☐ Editoria/Grafica/Pubblicità
- **Q** ☐ Pubblica amministrazione centrale/Locale
- **R** ☐ Consulenza legale/Commerciale
- **S** ☐ Commercio/Distribuzione
- **T** ☐ Istruzione (Scuola/Università)
- **U** ☐ Formazione/Ricerca
- **V** ☐ Broadcast/Audio e video professionale
- **Z** ☐ Strumenti musicali
- **X** ☐ Altro (specificare) ______

N. DI DIPENDENTI
- **A** ☐ da 1 a 49 **C** ☐ da 250 a 999
- **B** ☐ da 50 a 249 **D** ☐ da 1000 in su

FUNZIONI
- **AA** ☐ Acquisti
- **BB** ☐ Vendite
- **CC** ☐ Progettazione/Ricerca e sviluppo
- **DD** ☐ Marketing e Comunicazione
- **EE** ☐ Direzione Generale
- **FF** ☐ Produzione
- **GG** ☐ Amministrazione/Personale/Finanza
- **HH** ☐ EDP
- **II** ☐ Professionista/Consulente
- **LL** ☐ Docente/Formatore
- **MM** ☐ Studente
- **NN** ☐ Altro (specificare) ______

INTERESSI PRINCIPALI
- **01** ☐ EDP
- **02** ☐ Personal Computer
- **03** ☐ Home Computer
- **04** ☐ Automazione Industriale e Meccanica
- **05** ☐ Strumentazione elettronica
- **06** ☐ Telecomunicazioni e telefonia
- **07** ☐ Elettronica professionale
- **08** ☐ Elettronica hobbystica
- **09** ☐ Elettrotecnica e impianti elettrici
- **10** ☐ Strumenti musicali
- **11** ☐ Marketing e management
- **12** ☐ Broadcast/audio e video professionale
- **13** ☐ Didattica
- **14** ☐ Altro (specificare) ______

CHE PERSONAL COMPUTER POSSIEDE
- **DOS** ☐ MS DOS e compatibili
- **MAC** ☐ Macintosh
- **AMG** ☐ Amiga
- **C64** ☐ Commodore 64
- **VAR** ☐ Altro home computer

COMPILARE E SPEDIRE IN BUSTA CHIUSA A: GRUPPO EDITORIALE JACKSON - VIA ROSELLINI 12 - 20124 MILANO

SERVIZIO QUALIFICAZIONE LETTORI

ABBONAMENTO GRATUITO
A 6 NUMERI, A SCELTA TRA LE SEGUENTI RIVISTE SETTIMANALI
☐ EO News Sett. ☐ INFORMATICA Oggi Sett. ☐ MECCANICA Oggi (da febbraio '89)

BARRARE LA CASELLA RELATIVA ALLA RIVISTA PRESCELTA

INDIRIZZO PRIVATO

COGNOME E NOME ______

VIA E NUMERO ______

CAP ______ CITTÀ ______ PROV. ______

TEL. (______) ______ ANNO DI NASCITA 19 ______

TITOLO DI STUDIO: ☐ LAUREA ☐ MEDIA SUPERIORE ☐ MEDIA INFERIORE

INDIRIZZO LAVORO

DITTA O ENTE ______

VIA E NUMERO ______

CAP ______ CITTÀ ______ PROV. ______

TEL. (______) ______ TELEX ______

ATTIVITÀ AZIENDA
- **A** ☐ Informatica
- **B** ☐ Automazione Industriale
- **C** ☐ Meccanica
- **D** ☐ Strumentazione elettronica
- **E** ☐ Telecomunicazioni e Telefonia
- **F** ☐ Elettronica
- **G** ☐ Chimica
- **H** ☐ Elettrotecnica e Impianti elettrici
- **I** ☐ Laboratori di analisi Chimica e medica
- **L** ☐ Altra industria manifatturiera
- **M** ☐ Agricoltura
- **N** ☐ Ingegneria/Edilizia/Architettura
- **O** ☐ Finanza/Banche/Assicurazioni
- **P** ☐ Editoria/Grafica/Pubblicità
- **Q** ☐ Pubblica amministrazione centrale/Locale
- **R** ☐ Consulenza legale/Commerciale
- **S** ☐ Commercio/Distribuzione
- **T** ☐ Istruzione (Scuola/Università)
- **U** ☐ Formazione/Ricerca
- **V** ☐ Broadcast/Audio e video professionale
- **Z** ☐ Strumenti musicali
- **X** ☐ Altro (specificare) ______

N. DI DIPENDENTI
- **A** ☐ da 1 a 49 **C** ☐ da 250 a 999
- **B** ☐ da 50 a 249 **D** ☐ da 1000 in su

FUNZIONI
- **AA** ☐ Acquisti
- **BB** ☐ Vendite
- **CC** ☐ Progettazione/Ricerca e sviluppo
- **DD** ☐ Marketing e Comunicazione
- **EE** ☐ Direzione Generale
- **FF** ☐ Produzione
- **GG** ☐ Amministrazione/Personale/Finanza
- **HH** ☐ EDP
- **II** ☐ Professionista/Consulente
- **LL** ☐ Docente/Formatore
- **MM** ☐ Studente
- **NN** ☐ Altro (specificare) ______

INTERESSI PRINCIPALI
- **01** ☐ EDP
- **02** ☐ Personal Computer
- **03** ☐ Home Computer
- **04** ☐ Automazione Industriale e Meccanica
- **05** ☐ Strumentazione elettronica
- **06** ☐ Telecomunicazioni e telefonia
- **07** ☐ Elettronica professionale
- **08** ☐ Elettronica hobbystica
- **09** ☐ Elettrotecnica e impianti elettrici
- **10** ☐ Strumenti musicali
- **11** ☐ Marketing e management
- **12** ☐ Broadcast/audio e video professionale
- **13** ☐ Didattica
- **14** ☐ Altro (specificare) ______

CHE PERSONAL COMPUTER POSSIEDE
- **DOS** ☐ MS DOS e compatibili
- **MAC** ☐ Macintosh
- **AMG** ☐ Amiga
- **C64** ☐ Commodore 64
- **VAR** ☐ Altro home computer

COMPILARE E SPEDIRE IN BUSTA CHIUSA A: GRUPPO EDITORIALE JACKSON - VIA ROSELLINI 12 - 20124 MILANO

SERVIZIO QUALIFICAZIONE LETTORI

ABBONAMENTO GRATUITO
A 6 NUMERI, A SCELTA TRA LE SEGUENTI RIVISTE SETTIMANALI
☐ EO News Sett. ☐ INFORMATICA Oggi Sett. ☐ MECCANICA Oggi (da febbraio '89)

BARRARE LA CASELLA RELATIVA ALLA RIVISTA PRESCELTA

INDIRIZZO PRIVATO

COGNOME E NOME ______

VIA E NUMERO ______

CAP ______ CITTÀ ______ PROV. ______

TEL. (______) ______ ANNO DI NASCITA 19 ______

TITOLO DI STUDIO: ☐ LAUREA ☐ MEDIA SUPERIORE ☐ MEDIA INFERIORE

INDIRIZZO LAVORO

DITTA O ENTE ______

VIA E NUMERO ______

CAP ______ CITTÀ ______ PROV. ______

TEL. (______) ______ TELEX ______

ATTIVITÀ AZIENDA
- **A** ☐ Informatica
- **B** ☐ Automazione Industriale
- **C** ☐ Meccanica
- **D** ☐ Strumentazione elettronica
- **E** ☐ Telecomunicazioni e Telefonia
- **F** ☐ Elettronica
- **G** ☐ Chimica
- **H** ☐ Elettrotecnica e Impianti elettrici
- **I** ☐ Laboratori di analisi Chimica e medica
- **L** ☐ Altra industria manifatturiera
- **M** ☐ Agricoltura
- **N** ☐ Ingegneria/Edilizia/Architettura
- **O** ☐ Finanza/Banche/Assicurazioni
- **P** ☐ Editoria/Grafica/Pubblicità
- **Q** ☐ Pubblica amministrazione centrale/Locale
- **R** ☐ Consulenza legale/Commerciale
- **S** ☐ Commercio/Distribuzione
- **T** ☐ Istruzione (Scuola/Università)
- **U** ☐ Formazione/Ricerca
- **V** ☐ Broadcast/Audio e video professionale
- **Z** ☐ Strumenti musicali
- **X** ☐ Altro (specificare) ______

N. DI DIPENDENTI
- **A** ☐ da 1 a 49 **C** ☐ da 250 a 999
- **B** ☐ da 50 a 249 **D** ☐ da 1000 in su

FUNZIONI
- **AA** ☐ Acquisti
- **BB** ☐ Vendite
- **CC** ☐ Progettazione/Ricerca e sviluppo
- **DD** ☐ Marketing e Comunicazione
- **EE** ☐ Direzione Generale
- **FF** ☐ Produzione
- **GG** ☐ Amministrazione/Personale/Finanza
- **HH** ☐ EDP
- **II** ☐ Professionista/Consulente
- **LL** ☐ Docente/Formatore
- **MM** ☐ Studente
- **NN** ☐ Altro (specificare) ______

INTERESSI PRINCIPALI
- **01** ☐ EDP
- **02** ☐ Personal Computer
- **03** ☐ Home Computer
- **04** ☐ Automazione Industriale e Meccanica
- **05** ☐ Strumentazione elettronica
- **06** ☐ Telecomunicazioni e telefonia
- **07** ☐ Elettronica professionale
- **08** ☐ Elettronica hobbystica
- **09** ☐ Elettrotecnica e impianti elettrici
- **10** ☐ Strumenti musicali
- **11** ☐ Marketing e management
- **12** ☐ Broadcast/audio e video professionale
- **13** ☐ Didattica
- **14** ☐ Altro (specificare) ______

CHE PERSONAL COMPUTER POSSIEDE
- **DOS** ☐ MS DOS e compatibili
- **MAC** ☐ Macintosh
- **AMG** ☐ Amiga
- **C64** ☐ Commodore 64
- **VAR** ☐ Altro home computer

GRUPPO EDITORIALE
JACKSON
ED E' SUBITO MUSICA
CON AMIGA
CORSO RAPIDO,
FACILE, COMPLETO
NUMERO UNICO
A sole £ 19.000
Frs. 28,50
MUSICA
con
AMIGA
GRUPPO EDITORIALE
JACKSON
MUSICA
con
AMIGA
IN EDICOLA